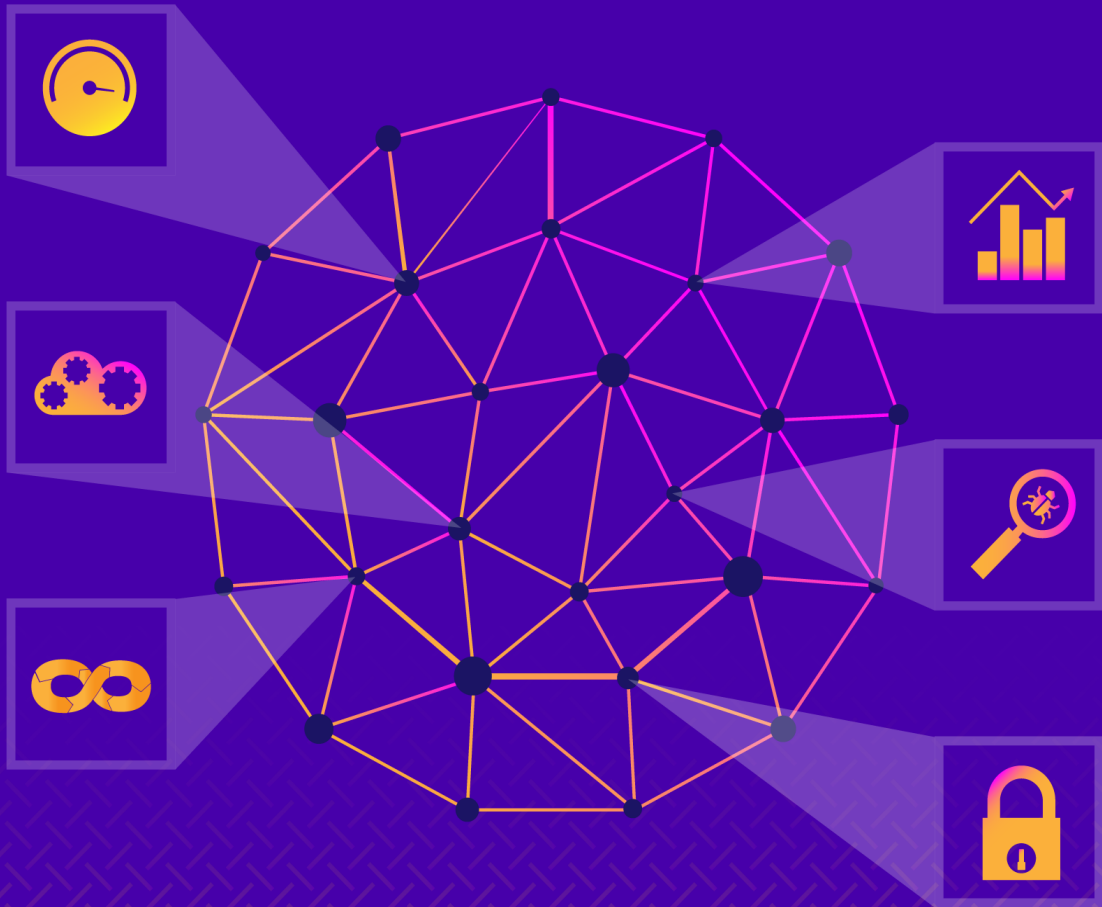


The API Economy

Disruption and the Business of APIs



AUTHORS

Bill Doerrfeld
Chris Wood
Art Anthony

Kristopher Sandoval
Arnaud Lauret



NORDIC APIS
nordicapis.com

The API Economy

Disruption and the Business of APIs

Nordic APIs

© 2015 - 2016 Nordic APIs

Contents

Preface: Introducing the API Economy and the Business of APIs	i
Tracking the Growth of the API Economy	1
Growth by the Numbers	2
The Birth of B2D	3
Growth by Sectors	5
Analysis: Evidence of Market Maturity	10
11 New Breeds of Businesses That Have Emerged out of the API Economy	12
1: Company that Provides API to Developer Consumers as Top Priority	13
2: Company Whose Infrastructure Depends Heavily on Third Party APIs for Survival	14
3: Company that Supports APIs with Management Solutions	14
4: Companies that Specialize in API Testing and Monitoring	15
5: API Documentation-as-a-Service	17
6: Tools for API Development and Continuous Delivery	17
7: Cloud Hosting	18
8: Companies that Specialize in API Discovery and Marketing	19
9: API Brokers - Independent API Strategy Consultants	20
10: Companies that Bring APIs to the Laymen	21

CONTENTS

11: Organizations that Disseminate API Knowledge . . .	21
Analysis: More Innovative Business Types	22
Sectors for Exploitation with APIs	23
Cost Cutting	24
Food Data and Analytics	25
Freight Trucking	26
Green Building	27
Home Health Care	28
Analysis: Final Thoughts	30
The Role of APIs in Growing Financial Technology	31
FinTech and APIs: Making the Bank Programmable . . .	31
The State of FinTech	32
Advantages of Exposing a Bank with an API	33
In-Account App Marketplaces, other Proof of Concepts .	35
With PSD2, EU Banks Will be Open by Law	37
Regulatory Impact on Personal Data Rights	37
The Open Banking (and Data) Landscape	38
Analysis: Implications of Open Banking	42
How APIs are Streamlining Healthcare	44
Why Healthcare Needs APIs	45
Using APIs for Diagnosis	46
Improving Clarity for Patients	47
Making Operations Easier for Health Professionals . . .	48
The Trouble with Using APIs in a Healthcare Setting . .	49
Making Things Easier for Researchers	50
The Future of Healthcare and APIs	52
APIs Are Evolving The B2B Landscape	54
In With The New, Out With (Some) Old	55
The Clothing Chain Example	56
B2B 15 Years Ago: The Progression	56
Why B2B Strategy Really Needs An API Strategy	57
Getting The Conversation Rolling	58

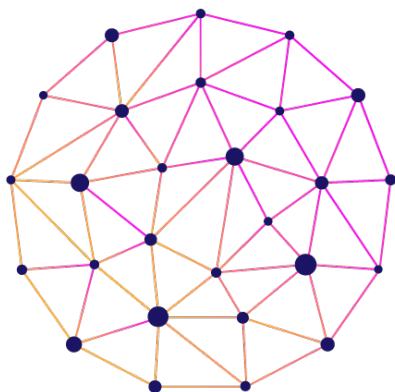
CONTENTS

API Advocacy	59
Convincing An Architect	59
Analysis: B2B World is Slowly Changing with APIs . . .	63
The Core Principles of API Management	64
A Definition of API Management	65
Components	67
API Registry	68
API Gateway	69
Developer Portal	71
Analysis: An Attempt at an Objective, General Definition	73
How APIs Are Disrupting The Way We Think	74
Disruption or Destruction?	75
The Biggest Disruption of All: APIs Change Our Expectations	77
APIs Are the Digital Glue Holding Our World Together	79
How Can Today's Developers Take Advantage of API Disruption?	80
Analysis: Drive API Disruption	82
Should Every Company Consider Providing an API? . . .	83
Common Myths Surrounding APIs	84
Myth #1: All APIs Are Public	84
Myth #2: Providing APIs Is an IT-Only Matter	84
Myth #3: Providing APIs Is Only For Internet Startups And Giants	85
Developing an Internal API: A First Crucial Step With Many Benefits	87
From The IT Department Perspective	87
From the Company Perspective	89
Providing APIs to Others: The Next Step	92
Analysis: APIs <i>Must</i> Be a Concern For Every Company	94
The Future of the API Space	95

CONTENTS

API Indicator #1: API Becomes an Accepted Tool to Grow Your Business	96
API Indicator #2: Microservices Architecture Allows All Departments to Scale	98
API Indicator #3: Are Public APIs Losing Their Luster? .	100
Analysis: Are You Getting All You Can out of the API Economy?	103
Endnotes	107

Preface: Introducing the API Economy and the Business of APIs



In 2006, the most predominant form of digital social communication was still email, and AOL instant messenger. A decade later, things have obviously changed quite rapidly in the face of higher bandwidth, greater capabilities, and the explosion of social media. Software-as-a-Service is an area that is growing exponentially. The digital platformification of older industries, paired with new advances in the Internet of Things (IoT) makes the API space — the powerhouse driving Internet connectivity — ripe for investment.

APIs, or Application Programming Interfaces, are an important cog in this process, and the market surrounding them is thriving. As John Musser of API Science told us, their future ubiquity throughout our digital fabric is inevitable. APIs encourage stan-

standardization — have you ever used Twitter to log in to a third party application? They **extend functionality** so that more potential is at our fingertips — how often do you query a map embedded into a web application? By exposing assets to developers to create new apps with, APIs also inspire **innovation**, promote data matter experts, lead to creative projects, and subtly increase the end user's experience.

APIs, in the family of SaaS (Software-as-a-Service) products, are part of an agile business methodology, which for the provider, can create entirely new revenue streams. Also an accelerator for corporate communication — the industry routinely uses Amazon Web Services as a hallmark of internal API adoption and the efficiency benefits therein.

APIs can be used to open new monetization streams alongside existing ones, but **API-first** companies have emerged that are entirely built around an API service. Twillio, Algolia, Contentful.com, and others are examples of companies that are exposing an API as their main product. But as John Donne says “No man is an island.” Every product we purchase is the culmination of a *lot* of group effort. Even something as simple as an apple is full of dependencies. A grower must rely on the seeds to sprout, purchase fertilizer, buy from land owners, hire pickers and distributors, work with retailers, and pay utility providers — all to produce a single fruit that the end consumer haphazardly throws into a shopping cart.

The API space has produced an economy in it's own right. There is obviously the monetary exchange from developer user to API provider. But looking deeper, **partners** to API programs have emerged that offer key components to support an API throughout it's lifecycle, streamlining operations for the API provider and improving their overall developer program. Whether it be marketing, development, documentation, metric analysis, security, and more, new microservices and platforms are available to smooth an API's creation and maintenance. This additional layer is strong evidence

of an industry maturing.

If you are a programmer looking for some code samples, this eBook may not be for you. This is the API Economy, the business of APIs. We'll forecast into the future growth of the API industry, analyze the impact APIs have historically had on specific industries, and see where existing APIs as well as partners to API providers fit into the overall spectrum. Something really cool that this volume explores are specific sectors that have not yet been API-fied (entrepreneurs — hint hint).

If you are an economist, industry expert, entrepreneur looking for a new project, or simply intrigued by the concept of disruptive APIs, read on...

Please enjoy *The API Economy*, and let us know how we can improve. Be sure to join the Nordic APIs [newsletter](#) for updates, and follow us for news on upcoming [events](#).

Thank you for reading!

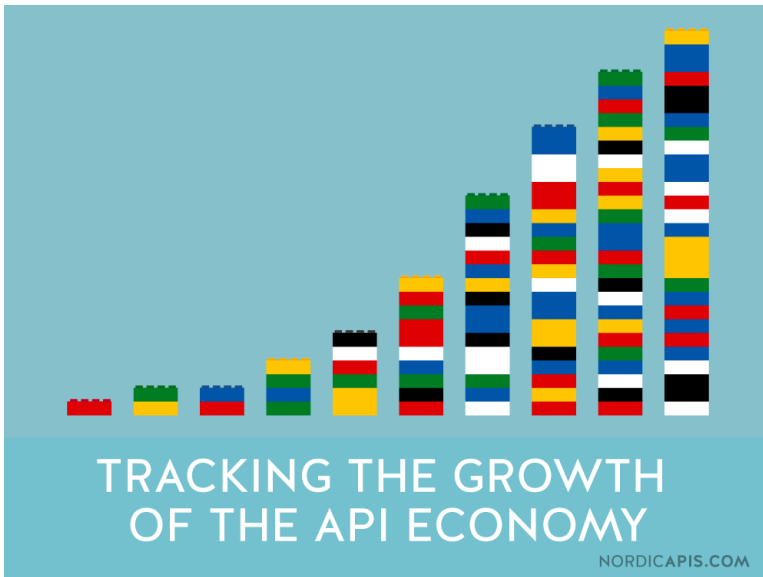
– Bill Doerrfeld, Editor in Chief, Nordic APIs

Connect with Nordic APIs:

[Facebook](#) | [Twitter](#) | [Linkedin](#) | [Google+](#) | [YouTube](#)

[Blog](#) | [Home](#) | [Newsletter](#) | [Contact](#)

Tracking the Growth of the API Economy



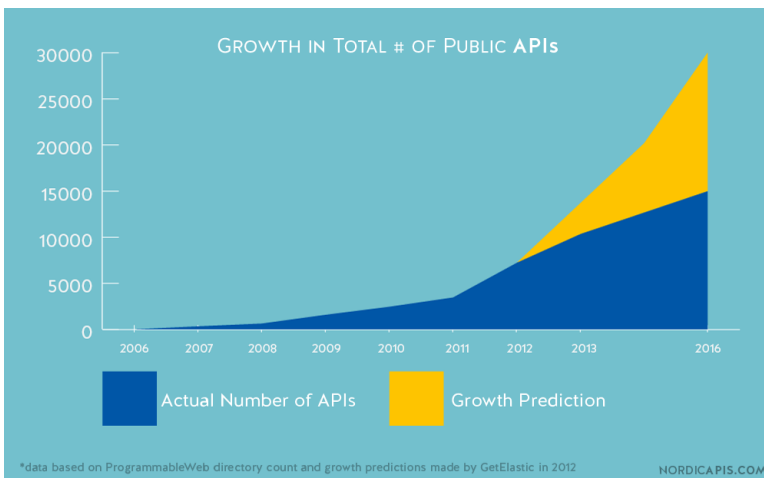
APIs are nothing new. Salesforce and eBay first allowed access to their web APIs in the year 2000, and other organizations were tinkering with the idea of exposing endpoints even before that.

The past few years, however, have seen such explosive growth that the **API space** is evolving more rapidly than ever before. Like “big data” and “the cloud” did before them, APIs are enjoying mass exposure and appeal to individuals beyond core groups of developers who deal with them on a daily basis.

But how does this growth compare with forecasts a few years ago? Is it sustainable? And why exactly has the space grown so rapidly?

Growth by the Numbers

Back in 2012, GetElastic and Monetate published an infographic that estimated that we would see 30,000 APIs by 2016, with 100 new APIs coming online each week. As it turns out, these estimates were a little on the high side. The ProgrammableWeb directory of APIs reported that in 2015, just shy of 2,000 APIs were [added to the site](#). This works out to around 40 APIs being added per week, and the total number of APIs represented on the directory currently stands at around 15,000.



Note that these numbers may be deceptive for quite a few reasons. First, there are now other API directories that need to be taken into account. For example, APIhound estimates there are 50,000 public web APIs, and APIs.io tracks over 1,000. Although there is certainly much overlap, some of these APIs may not yet reside in the manually curated PW.com directory.

More importantly, none of these numbers take into consideration the wealth of private or partner APIst that exist, which some estimate may even outnumber the public total. Some organizations

only bundle access to their API through [premium accounts](#), effectively making them less visible. These may not always be listed on app marketplaces because information about them is scarce or actively kept hidden, a discrepancy that [Craig Burton has identified](#):

“The key thing to consider here is that these numbers are based on publicly available APIs and do not reflect any private API growth at all... In all likelihood, any glitches that we see in Open API growth are expected to happen as the private sector catches up or even surpasses Open API growth.”

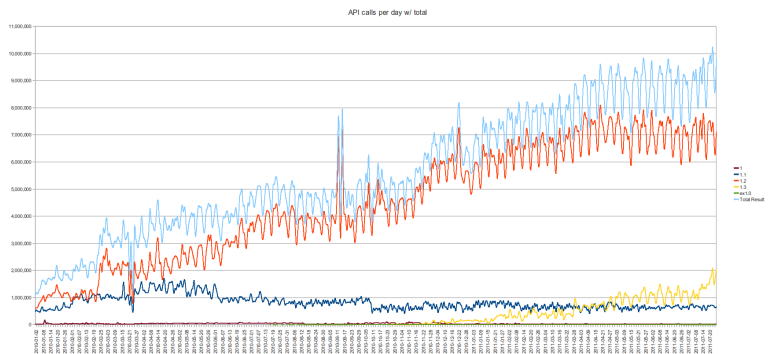
Another thing that early projections may not have taken into account is that so many developers would trust consuming [third party APIs](#) rather than building internal solutions themselves. For example, there’s only one MailChimp API but ProgrammableWeb cites 15 mashups that have been built using it.

The Birth of B2D

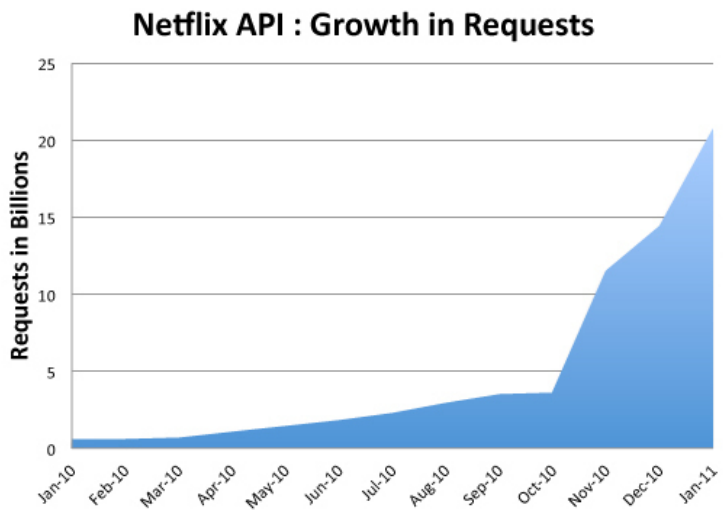
Writing for the Graydon blog, Alice Payne [posits](#) B2D, or “business to developer”, as a new form of marketing that has emerged primarily as a result of APIs.

If you’re reading this post, you’re undoubtedly already aware that APIs are big business. But what you may not know, and what many early API growth projections did not necessarily take into account, is just how **dominant** some APIs are in their spaces today.

Look at the rate of API calls for any popular API in recent years and you’ll see a graph shape that displays rapid growth:



Source: [MailChimp](#)



Source: [Netflix](#)

The Netflix API has since turned private so there’s no fresh data to compare, but MailChimp more than [quintupled](#) the amount of API requests they handle each day to more than 50 million by 2015 and 80 million by the [beginning of 2016](#). They also have a development fund worth 1 million dollars devoted to encouraging up and coming

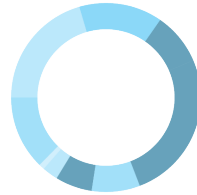
devs to build using their API.

Given all of this, it's easy to see why so many devs are compelled to use dominant APIs that are already out there and have established a good reputation rather than trusting a newer, smaller API provider. On the other side of the coin, this also accounts for why some would-be API providers haven't bothered trying to compete with the big players in the market.

Essentially, emerging monopolies in the space could account for a **levelling effect** of the exponential growth ratio of APIs, but doesn't at all translate to a capping of actual total API calls.

Growth by Sectors

The growth of the API economy isn't due to any one factor, rather a perfect storm of sectors emerging that all rely on APIs to some degree. Throughout this eBook we will explore various sectors impacted by the API revolution, but for now we'll take a look at a few of the major ones, which are particularly important because they've helped to demonstrate the usefulness of APIs to the general laymen population.



API consumer market

APIs are no longer viewed as complicated, stuffy services that only experienced developers can use. Apps like [IFTTT](#) and [Zapier](#) are interesting because they've opened up the functionality of APIs to non-devs. If you've ever seen someone cross-posting on Instagram, Facebook, and Twitter simultaneously then you've likely seen IFTTT in action.

While it's true that many [hugely popular apps](#) make use of APIs, they usually do so in such a way that the average consumer probably doesn't realise it. Zapier and IFTTT are among the first to transparently use APIs in a way that the average Internet user can understand.

In the same way that Myspace enabled a generation of amateur coders by allowing the use of HTML to create custom profile pages, IFTTT and Zapier are creating amateur API consumers by encouraging them to create their own recipes or Zaps.

What makes this particularly important is that it's going on at the same time as the emergence of the [Internet of Things](#).

IoT

The Internet of Things is a notable use for APIs because it represents another area in which the general public will become more familiar with the inner workings of APIs. We've previously [written at length](#) on how APIs drive the IoT, and [others have too](#), but it bears repeating just how important the relationship is, especially as IoT design is still evolving.

Time will tell exactly how prevalent IoT enabled devices become, but the rapid adoption of smart wearables hooked up to Nike+, Amazon's Echo, and so on suggests that there's definitely an audience for the sort of hyperconnected world that the Internet of Things promises. In fact, 44% of API providers [surveyed by Smartbear](#) in 2016 believe that IoT will drive the most API growth in the next two years.

FinTech

Finance is typically an area that has been very slow to warm up to new technology, sticking with tried and tested methods that have been around for years. However, enough banks and financial

institutions have embraced change that there are some exciting developments in FinTech.

APIs are changing the face of [personal banking](#) – as well as that of the [financial space](#) more generally – with quicker onboarding processes, partnerships and better integration with other services representing just a few of the changes that have taken place in recent years.

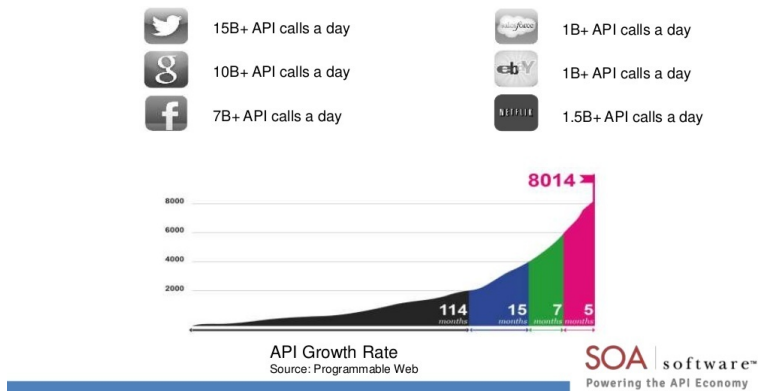
APIs also have a vital role to play in the world of cryptocurrency, such as with [Block.io](#), who use APIs to provide a wallet that can store litecoin, dogecoin and bitcoin in one place, as well as enabling users to build apps to process transactions, forward payments and create different types of secure wallets.

Social Networks

There are billions of smartphones in [use across the world](#), with over 200 million in the U.S. alone, and countless mobile services are built on APIs to the extent that they rely on them to function properly.

The data below is a few years old but demonstrates, even then, how important APIs are to the social space.

API Frenzy



Source: [API Frenzy – SlideShare](http://www.slideshare.net/SOA_Software/api-frenzy-api-strategy-101)

Apps like iAlien and TweetDeck, mobile clients for Reddit and Twitter, only came into existence thanks to open APIs from the two sites. Reddit didn't even offer an official mobile app for a long time, and TweetDeck was ultimately **acquired by Twitter in 2011** after being identified as a key factor in the company's mobile growth.

As social networks continue to appear, grow and evolve, third party developers and APIs will continue to play a vital role in the process of expansion via mobile and desktop apps.

B2B Products and Services

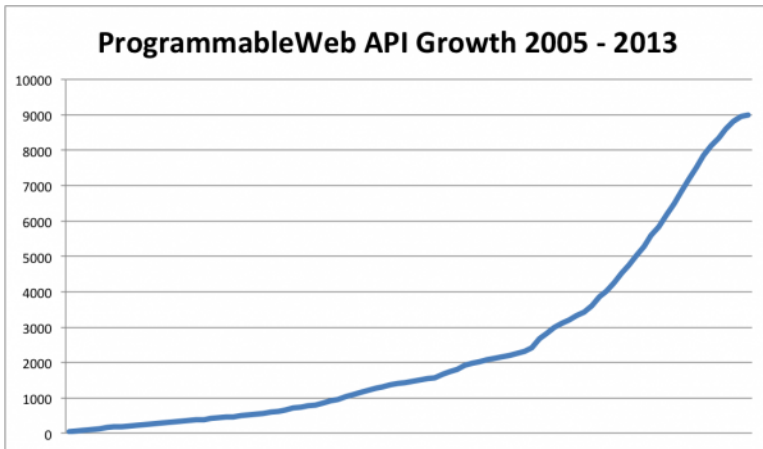
The **B2B space** is another area in which APIs continue to have a significant impact. A common growth technique for startups and other growing companies is to create integrations with other organisations, either by manually linking up their APIs or through a third party like OneSaaS, often with the aim of reducing the amount of time that has to be spent on entering data more than once.

Another thing to consider is the way in which some organizations include access to their API as a perk to customers above certain price points. For example, Salesforce [allows integration](#) via web service API only to Enterprise and Unlimited customers. It's worth noting that they generate 50% of their revenue [through APIs](#).

Integration, and thus APIs as well, continues to be a key factor for B2B organizations and the idea of "connectedness" is one that persists. 41.4% of those [surveyed by Smartbear](#) responded that integration with existing tools is the first thing they look at when assessing the potential for using an API, and 39% cite better interaction between products/services they use as the second most important issue they deal with when using tools in the workplace.

Analysis: Evidence of Market Maturity

Some of the numbers above suggest that API growth is slowing down, but this is nowhere near true to the degree that you might deduce.



Source: [ProgrammableWeb](#)

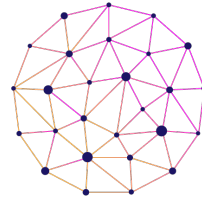
Notice the way that the above graph, which looked exponential in 2013, starts to level out towards the end. This is a trend that continued into 2014 and into 2016, but it could actually be argued that this is a good thing.

“Hockey stick growth” is often unsustainable and, in the case of APIs, might indicate a glut of subpar or unreliable products. Given the importance of uptime and scalability in the API space, that could be damaging to the reputation of APIs as a whole.

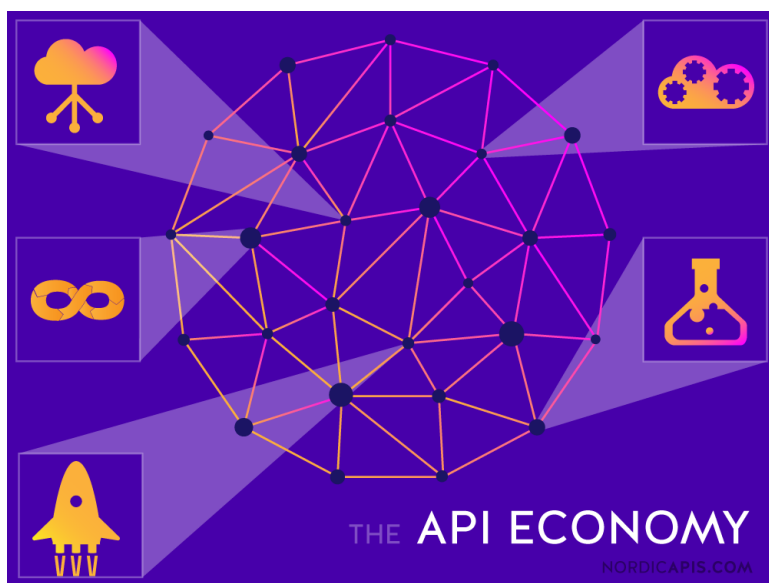
It could be said that the API space is maturing, growing horizontally rather than vertically. The trend honeymoon, during which many organizations jumped on the open API bandwagon because it was seen as “the thing to do” ... only to abandon/retire them soon after, is over but the usage of APIs continues to explode.

The number of APIs may not be growing at the rate that was predicted a few years ago, though it's not too far off, but their usage and the [surrounding economy of services](#) shows no signs of stalling.

With the mainstreaming of APIs through popular services that make use of them, organisations making APIs a serious factor in their business model and the progress of mobile technology and the Internet of Things, there's little doubt that the space will continue to grow and flourish at a rapid pace. In the next chapter, we'll expand on the growth of services that support this new economy.



11 New Breeds of Businesses That Have Emerged out of the API Economy



Unprecedented tech revolution often spawns brand new services and companies. As technological evolution opens new possibilities for humanity, it also leads to a flood of ideas and new business ventures never before possible. Now, as hyper-specialization occurs throughout the Internet, the **API economy** has become a macrocosm in its own right, spawning brand new types of adjunct companies.

Once simply a strategic offshoot for existing tech, APIs or Application Programming Interfaces have transformed into Software-as-a-Service (SaaS), the bread and butter upon which many apps are nowadays built. With this reliance and wide spread use, innovative tooling and new industry experts have come onto the scene to support APIs.

A lot has been said about the emergence of the API economy/industry. But what exactly has emerged? What type of businesses are pioneering in this new ecosystem, and what tools or services do they contribute exactly? Thus, the goal of this chapter isn't to showcase certain providers, but rather to make sense of the ever growing industry through categorization. We've done our best to identify new breeds of companies that quite literally owe their existence to the emergence of the world of APIs.

1: Company that Provides API to Developer Consumers as Top Priority

For some time now, many companies have provided a public API as a means for others to access their infrastructure, but this was historically done on the sidelines. We now see more and more API-first companies emerge whose **entire business model** exists around developers using their [Software-as-a-Service](#). These sort of APIs are often niched, performing one functionality extraordinarily well, and monetized in a [freemium model](#), like the Stripe payment API, for example.

Companies have built and sold tooling — languages, frameworks, etc. — for web developers to create new software with for decades. This relationship between software vendors and developers isn't new, but the standardization of offerings and ubiquity in the API space especially has proven quite substantial.

2: Company Whose Infrastructure Depends Heavily on Third Party APIs for Survival

There are of course the users, the **consumers** of APIs. Many mobile apps are essentially married to **various APIs**. In our article on choosing **brick products**, we identify the emergence of the composite enterprise — the core IT for large industries is becoming an assemblage of **microservices** as well. Core company IT may be completely API-fied, as in the case with streamlining internal operations at Amazon or Netflix.



APIs are also leveraged to grow social networks — some have argued that without third party developer apps, explosive growth with **Instagram** for example, would not have been possible. From our research, it's safe to say that most digitally savvy companies now rely on either Private, Public, or **Partner APIs** in order function in the state we know them.

In a **composite enterprise**, a competitive advantage thus arises out of the unique assemblage of functionality. In the future, what companies bring to the table will become only increasingly embedded with the digital services at hand, thus increasing the reliance on near **100% uptime** for SaaS.

3: Company that Supports APIs with Management Solutions

Providing an API is no easy task. Even after the core functionality is programmed, you must still consider hosting, filtering API calls, monitoring usage, access control, documentation, how **DevOps** will handle ongoing maintenance, and much more.

Because of this, a sea of platforms has emerged to support APIs throughout their lifecycles, allowing developers and non developers alike to create, run, distribute, and/or monetize APIs. **API Management** may involve offering server space for API functionality, a front-end documentation for developer consumers, a dashboard for performance monitoring, billing, and/or specific security solutions. We won't attempt to list of all the types of solutions or providers available, but they include things like:

- Comprehensive API management solutions;
- monitoring tools for analytics;
- gateways, rate limiting
- Middleware from enterprise to cloud connect applications to integrate with new business partners
- key generation
- security tools for access management, identity control

'API management' is still a nebulous term, and we've attempted to reach a [vendor neutral definition](#) in the past that includes the **Developer Portal** — the forward facing facade that developers can visit to learn about your API, the **API Registry** — where data and functionality is actually stored, and the **API Gateway** — that which receives API calls and filters traffic. API management is certainly an area that has seen much expansion. API Evangelists lists [52 different management companies](#) in his research.

4: Companies that Specialize in API Testing and Monitoring

APIs need a careful eye. To avoid bugs, and to improve all around [Developer Experience](#), **testing** must be routinely performed, especially before deployment, but at all times for [continuous integration](#) scenarios.

Testing may be part of an API management solution, but [many niche companies](#) have emerged that specialize solely on testing, helping automate functionality testing and performance monitoring for your APIs or [API dependencies](#). These tools often act as a proxy which can perform things like:



- **load testing:** demand is put on the API to determine how it behaves under peak load conditions
- **performance monitoring:** helps identify bugs and makes sure the API behaves as expected, helps consumers adhere to their service level agreements
- **resource management:** the monitoring of resources that support API performance
- **track usage:** automated logging of use to track method access and analyze traffic

A testing paradigm could be manually developed as a simple tool to test requests, but as we've mentioned before, it doesn't always help to [reinvent the wheel](#). There are companies that go a little further with [virtualization](#), a front-end Testing-as-a-Service for your developer consumers that is "mockups on steroids" which mimicks the live API performance.

The ongoing monitoring of uptime and performance behavior is cornerstone to a quality API with good developer relations. Though outsourcing may not fit every situation, a well-polished automated monitoring mechanism means increased response time to errors, and better preparation for increased traffic, both meaning an increase in end user happiness.

5: API Documentation-as-a-Service

Some have suggested that in the near future you will literally own *nothing*. Instead, everything will be consumed in the same way we now use media or integrate cloud software components — as a subscription model. Not surprisingly, the Everything-as-a-Service (XaaS) trend spans to how we expose APIs as well.

The way we describe APIs is vital. A key component to the [developer hub](#), **documentation** is the visual, forward facing resource that users will refer to until the end of your service's life cycle. Great documentation is often structured in the [3 columned approach](#), outlining all potential requests and functions, the proper HTTP calls to use, and code examples.

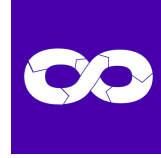
Documentation should be readable and optimized to increase the onboarding time for the service — this means lucid description, ample commenting, excellent architecture, layout design, and more. With good reference being such a fundamental tool for API existence, it's no wonder that experts have arisen, and that providers are willing to pay a per-month charge for their services.

Tools like [Swagger](#), owned by SmartBear and powering the Open API Initiative, can be used to generate API server code, client code, and documentation for these services. Or take [Readme.io](#), which can be used to create API references, extend a developer portal with a platform for additional guides, code packages, use cases, tutorials, common troubleshooting questions, and more.

Though some [API specification formats are going open source](#), these projects are supported by parent companies with a claim stake in supporting the API economy.

6: Tools for API Development and Continuous Delivery

The wealth of companies creating tools for building out and maintaining APIs is tremendous — making it very difficult to sum up in a single section. To get an idea, subcategories include:



- **Web frameworks:** Microframeworks specifically designed for building lightweight web services are all the rage. We've experimented with using [Spark](#), [Play](#), and [Lumen](#) frameworks in the past.
- **Continuous integration:** Rapid change requires iterative and quick releases. Therefore, tools that support the [DevOps](#) approach to API development are numerous.
- **Configuration management.** Docker containerization has changed the way the build process for web development is packaged.
- **Library generation:** Tooling has emerged that enables one to automatically derive SDKs or language specific code libraries from their API specification.

7: Cloud Hosting

Every web application needs a place to live. Cloud computing didn't exactly emerge *out of* the API economy like others on this list, but it is a huge part of the API industry equation — the thousands of chugging APIs have certainly increased our reliance on cloud server providers.

In the 90s and 2000s choosing a service to host a PHP application, for example, was relatively straightforward. Today, many frameworks exist — it seems that every cloud PaaS provider is doing something slightly different. Writing for [Airpair](#), Daniel Rice outlines how to go about hosting a [Ruby on Rails](#) application on various cloud

computing providers. These service charge monthly fees for varying bandwidth, providing things like:

- a virtual server, permanent cloud storage for resources,
- using a console to create instances and deploy,
- aggregation of virtual private servers,
- and targeting for geographical regions.

A software's cloud stack will rely on the makeup of the technology from which the API is derived. Whether you go through Amazon Web Services, DigitalOcean, Heroku, or others, every web application needs to exist somewhere.

8: Companies that Specialize in API Discovery and Marketing

As a developer, finding an integration for an app may be as simple as a Google search. Nevertheless, many have sought to profit in some way from the overwhelming amount of APIs available by organizing them into **directories**, or designing new formats to automate search and discovery.



Many discovery tools — [we listed 11 ways to find APIs](#) — are available to help entrepreneurs and app developers discover APIs to integrate with. Some API management solutions supply publicly searchable directories, while other aggregations are manually curated or automatically compiled using unique methods.

For discovery by web search to be possible, typical SEO and digital marketing should be considered. But making something discoverable is really the tip of the iceberg when promoting technology; [developer engagement and evangelism are full time jobs](#).

This outreach and community-building aspect of API marketing is starting to be outsourced by niche agencies. For example, [Catchy](#) specifically help grow developer communities, a strong parallel with marketing API programs.

9: API Brokers - Independent API Strategy Consultants

With all these new businesses, novel technology, confusing terminology, and competing services, the industry is certainly daunting for newcomers. There is certainly room for experts to fill in the gaps. In 2014, Cyril Galliard described [the need for an API Broker](#) to consult his startup, stating that “Understanding the World of APIs” was the #1 important asset he could hope for in a co-founder. Similar thoughts were echoed throughout the blogosphere. API experts that follow the industry could aid businesses in the following ways:

- **Brokers** understand the economy and can recommend integrations,
- **Strategy consultants** review existing APIs, consult the overall business strategy
- **Programmers** are contracted to offer design advice, develop an API, or build libraries
- **API advocate for hire:** individuals who have experience in API marketing, evangelism, or outreach will become more and more valuable.

Already, many consultants like Mike Kelly ([Stateless.co](#)), James Higganbothan ([LaunchAny](#)), Kin Lane ([APIware](#)), ([Hitchhq](#)), and others have emerged that are doing this sort of personal consultation on API strategy, design, and implementation.

10: Companies that Bring APIs to the Laymen

New services like Zapier and IFTTT enable users to create their own functionality mashups using APIs. These Platform-as-a-service companies are very exciting to us, as they are some of the first services transparently offering plug and play API functionalities so that **non-devs** can now interact and take advantage of APIs.

11: Organizations that Disseminate API Knowledge

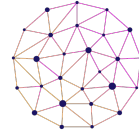
Last but not least on our list are the groups that spread the word of API. There are plenty of conferences, blogs and news channels that circulate industry expert opinions, as well as coding academies that turn novices into practitioners.

The increasing number of dedicated channels covering the API industry is a clear indicator that a thriving economy has formed.



Analysis: More Innovative Business Types

The API industry hasn't been around for all that long, but in its decade of active growth has spawned a new ecosystem of interconnected services, with unprecedented companies that owe their existence to the ubiquity of this tech. Emergence is clear, but the next question to be asked is how many of these companies are sustainable, and turning a profit. The API economy is a thriving and high-yield field, certainly with much room for growth.



There is inherently overlap between the businesses models mentioned above — testing and documentation are often parcels of holistic API management “solutions”; infrastructure and deployment specifics may be embedded into a holistic cloud server arrangement, etc. The high impact of APIs on our web usage makes it difficult to condense or categorize web economy; other business models that could be appended to our list are things like:

- IoT gadgets and supporting services
- SOA-to-cloud integration specialists
- Analytics firms that track trends throughout the developer economy
- and more...

What will the future API economy resemble? We will likely continue to see more and more experts in the field emerge who specialize in even finer details surrounding the [API lifecycle](#).

Sectors for Exploitation with APIs



In 2015 “there’s an API for that” became an oft-repeated phrase, entering the vernacular with the same frequency as the app-based equivalent. Fueling the usage of this phrase is an increasing number of organizations taking their products to market using a SaaS-first (Software-as-a-Service) approach, exposing an Application Programming Interface or API at the core of their offering. This growth has delivered breadth to the [API economy](#), with the majority of industries offering a burgeoning number of APIs fulfilling an ever greater number of use cases.

In the past we’ve covered how areas like [IoT](#), [FinTech](#), [health-care](#), [sustainability](#), and others have become more and more pro-

grammable. Despite this growth there are doubtless some industries and sectors where APIs are relatively underrepresented, which could be of specific interest to entrepreneurs looking for the next growth market. In this chapter we take a look at some of these areas of potential growth to see how they align to gaps in the API economy.

Cost Cutting



An interesting growth area is cost cutting, with IT Business Edge [citing](#) third-party administration and insurance claims adjusting having grown 7.7% between 2011 and 2016 with many insurance companies outsourcing work in an effort to cut costs. APIs could be used in a number of ways in this sector:

- APIs could be used to create a **marketplace** for such services, allowing freelancers to bid for and win jobs alongside the large corporations;
- They can also serve as an **integration mechanism**, allowing adjusted claims to be logged and retrieved by any consuming application;
- Finally, APIs have a role to play in the **automation** and digitization of workflow and processes.

There appears to be few API providers trying to grow a SaaS-first offering in this space. One example of an organization developing their API offering (although not from the SaaS perspective) is the [Ability Network](#), which markets their [API toolkit](#) as helping insurance companies who are looking for “more efficient and less costly ways to obtain the data they need”. With some creativity

entrepreneurs could expand this approach, providing a platform for claims handling that could integrate with existing insurance toolsets and applications via an API.

Food Data and Analytics



The idea of the connected refrigerator has been kicking around the Internet of Things (IoT) space for the last 20 years, but that does not belie the fact that food data and analytics is a sector in the food industry that is ripe for growth. Inc.com [highlights](#) the fact that only 0.14% of a \$21.5 trillion global

market accounts for the development of “software for food logistics, processing, and retail” (according to research by [Frost and Sullivan](#)).

There are many APIs in the food space already, but the majority of these focus on subjects such as online ordering, take out, restaurant guide and recipes; a human being is involved as either the initiator or consumer. Food data and analytics differs in that much of the activity involved is autonomous, helping those working in the kitchen with their activities without having to ask. Some of the uses of APIs for food data and analytics include the following:

- Clearly the **self-ordering refrigerator** would become a reality, and spawn new conversations between the refrigerator, the foodstuffs inside it, and the supply chain taking place via APIs;
- The autonomous supply chain for the connected refrigerator would undoubtedly lead to the creation of a new **market-place** for food where a refrigerator’s shopping list could be sourced from different suppliers at the best price, all facilitated by APIs. Moreover, this has the scope to integrate

with many other APIs, such as loyalty and coupons, group buying, reselling or donating surplus foodstuffs, and so on;

- Like the connected refrigerator, the IoT components of the “connected kitchen” would also rely on APIs for **inter-communication**. Sensors on devices like food processors, scales, and saucepans would have a method for communicating with online recipe books or reacting to different food weights or other conditions that may change cooking times or the proportion of different ingredients.

Many of the protagonists in this space, including companies like [Innit](#), are specifically interested in the kitchen equipment “there is clearly a relationship between these kitchen IoT devices and APIs. However, from a SaaS-first perspective the market leaders will be those that help facilitate the autonomous supply chain, and it remains to be seen who will be first to market in providing such services at scale.

Freight Trucking



Forbes highlights freight trucking as a major growth area for small business, [quoting](#) a 25% increase in sales for the 12 months ending May 31 2015, with speciality haulage such as tankers and refrigeration experiencing growth rates at nearly double those across small business in all industries.

The trucking industry is always looking for cost-cutting measures “whether it be [aerodynamic tire flaps](#) designed to cut wind resistance, or [partially self-driving trucks that save fuel economy up to 10% by platooning](#). But what can data and software contribute?

On first impressions it appears that freight trucking is underrepresented in the API economy: For example, a keyword search on the term “trucking” in the ProgrammableWeb [API Directory](#) returns only 18 APIs listed using this phrase. The use cases for APIs in this industry would include the following:

- Acquiring **space** on partially full trucks;
- **Reselling business** when demand exceeds capacity (especially important for small businesses who may need to subcontract work a times of peak demand);
- Providing a **marketplace** for purchasing the best possible rate for shipping good across a range of carriers (either aggregating multiple data sources and surfacing the data with an API or aggregating other APIs).

There are some established products in this industry such as [Project 44](#). Project 44 recognizes the value APIs can bring to the trucking industry and specifically describes their product as an Integration-Platform-as-Service (iPaaS), connecting shippers to carriers via APIs. Entrepreneurs entering this sector could clearly exploit a similar approach, delivering API-based SaaS-first solutions to the market.

Green Building

The construction of buildings with minimal environmental impact has become one of fastest growing industries in the USA according to Go Banking Rates, which [cites](#) a predicted \$303 million share of GDP for this sector in the period 2015 to 2018. Whilst “green” APIs are becoming more common, due partly to home automation initiatives that allow external control of heating and lighting systems (for example [ThinkEco](#)), APIs related to *green building itself* appear to be relatively sparse. Their application for green building projects could be significant:

- If governments or regulatory authorities provided services via APIs to **approve building proposals** (including design and type and quantity of materials) in an automated fashion, this could increase the rate at which green buildings are constructed and lower the barrier to entry. The approval process is typically manual (for example, in the UK a series of inspections and reports is required making it a labor-intensive process) and providing equivalent services wherever possible via APIs would introduce many efficiencies;
- APIs could help provide **ongoing incentives** to green building owners, with APIs providing the facilities for monitoring the continued “greenness” of buildings in return for tax breaks or similar from the government.



There are of course APIs that provide some of the functionality required to create SaaS-first APIs for green building; for example, [Brighter Planet](#) offers an API that allows the consumer to calculate the carbon footprint for a variety of activities. However, to offer a SaaS-first API the provider will clearly need

to develop a platform that will facilitate many more aspects of green building.

Home Health Care

Go Banking Rates also highlights home health care as being a fast growing industry, with a predicted annual growth of 2.6% each year in the years to 2022: this is almost certainly a consequence of an aging population, with a predicted 20% of the US population over 65 by 2040. [Health care APIs are already experiencing significant growth](#), but *home* health care appears to be a niche that is underrepresented.

Like green building, APIs could be important facilitators for this industry for a number of uses:

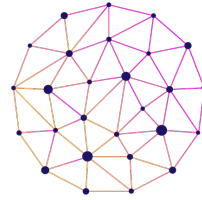
- Access to **patient records** for a distributed workforce of carers and health professionals will undoubtedly be critical to the growth of home health care. This is especially important for health sectors in countries like the USA where records could be held by multiple providers with many variations (as this [map](#) of patient record ownership shows);
- Like earlier examples, a **marketplace for caretaker services** could be established, fueling an Uber-like culture around sourcing and supplying care staff locally. Dovetailed with this approach is the regulation of staff, with APIs providing the means to perform due diligence and cross-checking of caretaker qualifications.



It stands to reason that governments will drive many [APIs in the health care space](#) (for example the US government's [HealthCare.gov developer portal](#) APIs), but clearly there is space for entrepreneurs to foster innovation through API-based services. Companies like [Validic](#), which provides a platform for connecting patients devices to health care providers are already making inroads with an API-based SaaS-first approach.

Analysis: Final Thoughts

There are many industries in the world that are growing rapidly which entrepreneurs will inevitably exploit. The examples we've investigated in this post highlight how a SaaS-first approach, fueled through APIs could bring even greater growth to these industries.



It is also important to consider that entering the API economy is a two-way street: APIs will directly benefit the stakeholders of these industries, but they can also bring those industries to new markets, with developers using APIs in innovative ways not conceived by the providers. If APIs fuel the further growth of these industries it will be an exciting time for the API economy as a whole.

The Role of APIs in Growing Financial Technology



FinTech and APIs: Making the Bank Programmable

Banking institutions are usually portrayed as monoliths, slow to change even amidst obsolescence and new technological advances. This won't be for long, as the **FinTech** sector is becoming increasingly open and programmable.

Even large U.S. banking institutions like [Citigroup](#), [BBVA Compass](#), [Bank of America](#) and [Capital One](#) acknowledge the benefits of opening internal systems for third party developers to integrate into

new apps. And, as alternative banking services like [Simple](#) or [Holvi](#) emerge, we are seeing more of a diversity of services that embrace new consumer expectations brought on by the Internet and mobile devices.

Through our research, we've tuned into FinTech experts to hear what organizations like **Fidor**, the **Open Bank Project**, and many others are doing to lead the banking revolution. Essentially, all agree that we're moving toward a new financial market of customer choice, third party services, and open data — all powered by APIs. Essentially this means **making the bank programmable**.

The State of FinTech

FinTech stands for **financial technology**; a broad sector of innovative and emerging financial services. Examples include crowd-funding platforms like [Kickstarter](#), new online-based currencies like [Bitcoin](#), virtual wallets, micro stock investment apps like [Robin-hood](#), account aggregation and analysis services like [Mint](#), payment splitting services like [Splitwise](#), new mobile-optimized peer-peer payment transfer apps like [Venmo](#) — the list goes on. Replacing physical charities, paper checks, and paper money, FinTech services are suitable to the digital times we are living in now, and are shaking existing banking infrastructure and international payment streams. Fintech craze as not simply a short lived phenomenon. The technologies being developed now are going to have **everlasting impacts** on the future financial and banking industry, and institutions need to adopt change or **face destruction**.

Advantages of Exposing a Bank with an API

Most banks lock customer data away in internal systems with very limited access, restricted to tightly controlled channels. This state of banking data accessibility is widely viewed by industry commentators and even banks themselves as nothing short of a travesty, epitomizing the hegemony banking giants have held for a long period of time. [According to Andres Wolberg-Stok](#), global head of emerging platforms and services at Citi, APIs present an opportunity to *“break a few windows to let free air and light in.”*

Whilst disrupters and innovators (especially challenger banks) will be prone to hyperbola, the common consensus is that banks should open up to provide APIs for a number of compelling reasons:

- **To enable consumer choice:** There are many consumers who want to be selective about banking products without having to choose a single banking provider, instead simply picking from a marketplace. APIs provide a mechanism to enable such selectivity of product. Supporting this notion is the great deal of coverage to the [“narrowing” of banking](#) (where the portfolio of bank activities is restricted), as well as the lack of trust millennials place in traditional banking establishments;
- **To unlock customer data:** Internet banking has become the key customer engagement point during the last two decades, but the digitization of the customer experience has made it clear that the vast majority of banks consider this data to be a vault with only one entrance and two sets of keys — theirs and the customers. This is juxtaposed with the wants and needs of customers, who are keen to exploit what is essentially their data in any number of different ways: Personal Financial Management (PFM), credit checks, digital notarization, and more. However, banks simply don’t provide

facilities for third-parties to work at the delegated authority of the customer, forcing many third party solutions to use the customer's internet banking login credentials and web scraping to function (examples include [Mint](#) and [Xero](#)). While this provides customers with useful solutions, by sharing their credentials with a third party they may actually break the terms and conditions of their bank's internet banking platforms;

- **To unlock themselves:** Anyone who has worked in the IT department of a large bank will be familiar with the architecture and approach that typifies them — extreme risk adversity with large amounts of governance on top of legacy systems and monolithic applications. An API-based architecture, built incrementally with many small steps could help unlock and decouple these architectures, making them more accessible and providing an environment to foster innovation.

The core theme is that banks could be doing much more to open customer data up to new use cases and business models. Naturally there are disrupters and innovators who are having some success trumpeting down the walls of Jericho, and the [Open Bank Project](#) is clearly the poster boy for the open banking movement. Another example is [Figo](#) in Germany that delivers a single API that integrates with the German FinTS/HBCI banking network. These initiatives show what can be achieved without a standardized API network, but a lot of hard yards are involved in creating the solutions with a myriad of different integrations across the banking ecosystem. Such efforts could be significantly reduced if each bank offered a standardized suite of APIs.

What banks need to create are well-designed, **standardized APIs**, along with self-serving adoption processes complete with documentation, sandbox, simulated account structures, and more to quickly get in the hands of developer users. Onboard more partnerships and make it cheaper for FinTech startups to launch, and you've got a

recipe for a banking API success.

In-Account App Marketplaces, other Proof of Concepts

A bank may choose to provide an API to allow developers to create add-on services in a **marketplace** format, enabling the end user to customize their banking experience. The idea is that a bank can provide an API and open app store (i.e. appstore.mybank.com) and allow users to pair certain apps with specific accounts. Win-win-win. Banks can acquire new partners, third party developers bring innovation, and customers are empowered with more choice and customization. It's true that for banks, building an application manager is no easy task, however, but **API management** solutions exist to aid this process.

Already, services already exist that are making waves in open banking and programmability:

[Open Banking Project](#) is a separate organization striving to reinvent how banks handle their data. The Open Bank Stack, written in [Scala and running on the JVM](#), is secured with [OAuth](#) and is a “semantic API vertical for the banking space”. They've had [numerous FinTech apps](#) developed that tap into their API service. Examples include [Savetastic](#), which pulls data from a bank to calculate potential savings, and [Social Finance application](#), which enables account users to choose who they want to share account data with.

New Platforms Lead to Unexpected Innovation

What's the similarity between a violin and a smartphone? In a session with Nordic APIs, Redfern, a composer at heart, argued that they're both **platforms**.

A violin and iPhone are both standardized interfaces. Yet, standardization doesn't inhibit musical creativity no more than the current smartphone design channels app developer curiosity and innovation. With platforms, weird things can happen, such as a rising class of Instagram entrepreneurs in Kuwait using the platform to [sell their sheep](#). In FinTech, the idea is that [an open standard API could revolutionize banking](#) with unprecedented consequences. Xignite CEO Stephane Dubois acknowledges that:

“The role of technology in advancing the financial service industry is more critical than ever before. The use of APIs by today's banks is becoming increasingly common as they help to drive speed and cost-effectiveness compared to traditional legacy systems.”

Just as Facebook and other social media giants have become a [platform](#), Redfern believes that **banks will inevitably become platforms as well**. The idea is that banks can push innovation and allow developers the ability to create brand new products, and the Open Bank Project can allow banks to easily adopt an API. To Redfern, open banking is made up of four distinct facets:

- Open standards
- Open Source
- Open data options
- Open innovation

“APIs are great for abstracting away aging IT systems which are barriers to innovation....We think banks need to open up their infrastructures, and they should be developing ecosystems so they can better respond to their customers and so they can compete better in the market”

It's been proven that customers are interested in using new entrant services rather than their own. According to the [Millennial Disruption Index](#), "71% would rather go to the dentist than listen to what banks are saying" — proof to some that the industry is ripe for "seismic" change.

With PSD2, EU Banks Will be Open by Law

On the European FinTech stage, forthcoming initiatives and regulations will both disrupt and foster innovation in the banking sector of the API economy. The most significant of these is the [Payment Services Directive 2](#) (PSD2), a new regulation that will apply across the European Union and is likely to result in a huge increase in the number of APIs for banking products. Making banks programmable will significantly change the engagement model for accessing a consumer's account. What is less clear is how this may affect the consumer themselves, including their level of access to the data (that *in theory* they own), and their ability to use their data in any way they see fit. Thus, how will PSD2 and the growth of APIs for banking affect personal data ownership?

Regulatory Impact on Personal Data Rights

The rationale for standard banking APIs is clear, and there has always been the potential for a large bank to "break cover" and offer a suite of APIs with access to customer data in advance of its rivals. However, in the absence of this early mover encouraging other banks to offer APIs by way of market competition, regulatory forces are now likely to coerce many into taking action.

PSD2 will force banks to allow third parties to access a given customer's data, where that third-party is acting as a data consumer or a delegated authority: The EU describes these as “[third party providers \(TPPs\) \[who\] offer specific payment solutions or services to customers](#)”. As this could include making a payment on a customer's behalf, PSD2 grants third-parties considerable power. There is a great deal of debate how this might be implemented from a technical perspective, but an obvious solution for anyone familiar with the API economy is the use of APIs to facilitate access. Moreover, APIs could be coupled with a rich framework such as [JSON Web Tokens under the guise of OpenID Connect](#) to provide strong authentication and non-repudiation.

It is an oversimplification to say using APIs and OpenID Connect would *immediately* provide the framework for implementing PSD2, and some of the logical architecture has already been framed: it includes the introduction of Account Information Service Providers that will provide a single view across multiple customer accounts. However, with the right governance framework these technologies could provide the bedrock of a new open banking landscape, supporting a wide range of new entrants to the market.

The Open Banking (and Data) Landscape

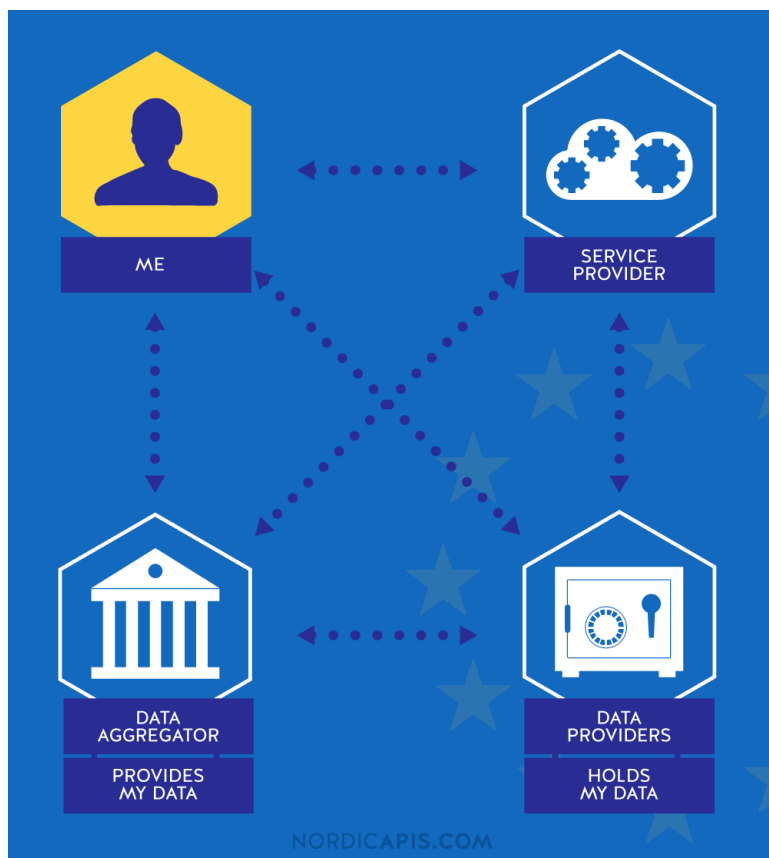
The open banking landscape will allow customers to unlock their data and give delegated authority to their payment instruments, empowering them to share it with whomever they saw fit via an API. The value in this unlocked data isn't restricted to banking — clearly other solution providers see much value in it. For example, frameworks like Figo will become much more commonplace with initiatives like PSD2 and possibly draw in data sources that aren't financial in nature, again via APIs.

Customers will have a single window on all their data, financial or not: while solutions already exist in this space, such as [Trunomi](#) and

[Mecco](#), who offer federated personal data stores and mechanisms to produce different views of an individuals' personal data, the integration effort is fraught with difficulty. With standardized APIs there is an opportunity to view, understand, and control a unified view of our data.

The consolidation of our digital persona into a single construct is both a risk and an opportunity for individuals: a risk, because without the right governance we offer ourselves up to solicitation on a huge scale, but an opportunity because we can enable the genuine usage of our data *for our own benefit*, truly empowering consumers to make insightful decisions. With such insight at our disposal concepts like the [Secco Aura](#) could become a reality for more people than just Secco bank customers, with a **broadcast of interests** possible across many different service providers, financial or otherwise. Such possibilities are so relevant in the data sphere that large organizations like [Visa Europe](#) are researching them, with their innovation lab currently running their “Me2B” theme. Howard Elsey, the innovation partner running this theme describes such a personal data network as:

“the connective tissue and nervous system of the data economy. This has ramifications into all current areas of technical innovation, from big data to blockchain, identity, and IOT but is so interesting ... because of the impact that it will have in overcoming something that computerization has managed lose — the personal nature and trust in the relationship between business and the individual and the disrespect of third party companies that exploit the value in an individual's data without returning value to the individual.”*

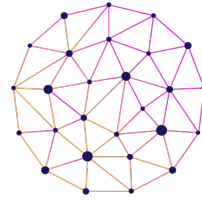


In architectural terms, APIs underpin this framework as both the personal data source, federated into what Mecco calls “the API of me” and the service delivery point. However, there is a new paradigm, the carrier for the broadcast of interests which, at face value looks and smells like massively distributed publish-subscribe network: the [NATS](#) technology for the personal data network. It is to this network that both consumers and providers broadcast their interests with APIs providing the delivery mechanism for data or services; requests can be serviced either directly from the consumer’s data stores or via aggregators that pull together and

consolidate the data in a myriad of different views. Some initiatives already exist that could grow into the backbone of this system, such as the [Open Mustard Seed](#) project, but however it comes about, the “network of me” will make the personal API economy truly revolutionary.

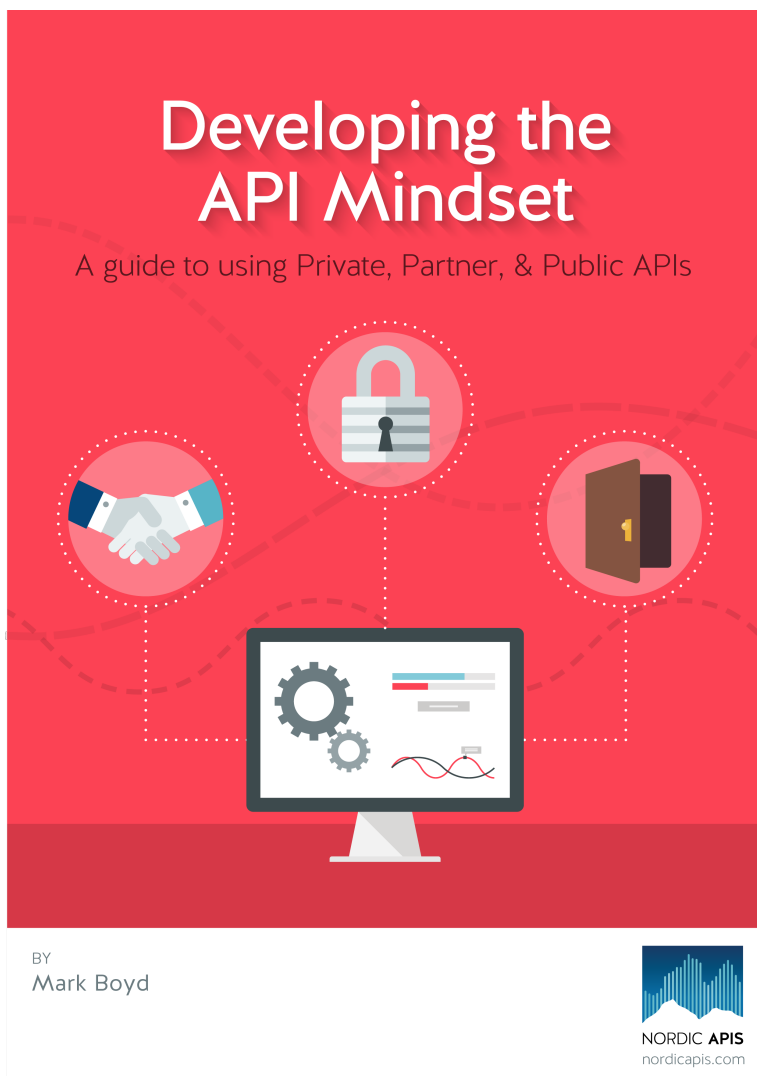
Analysis: Implications of Open Banking

The opening of banking through APIs, whether through competitive pressure or regulatory enforcement, has massive implications across the data ecosystem:



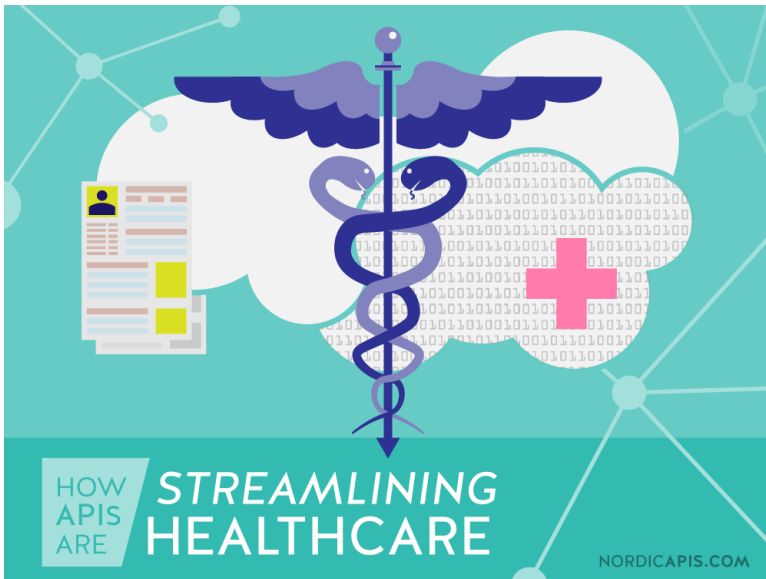
- **for consumers** it empowers their ability to access products and their data;
- **for solutions and services providers** it allows them to engage with customers in a much more seamless fashion;
- **for IoT** it provides the network for devices to make autonomous actions based on an individual's preferences, wants, and needs.

It will take work to come to fruition but these are exciting times for us all as consumers. It will be fascinating to see how banks receive the PSD2 regulation, and how this personal data network develops over the next couple of years. For developers, end users, and banks, **making the bank programmable is a win-win-win.**



For more on API business strategy check out [Developing The API Mindset](#)

How APIs are Streamlining Healthcare



Managing healthcare records can be a painful experience, for both patients and doctors. Though extensive medical records are kept on patients to be made available to new caregivers, accessing them is not always an easy process.

In the UK, records are held digitally (usually General Practice (GP) records) yet some are still written by hand (details of hospital visits, etc.). And just to make matters even more complicated, the same data types are at times held partly digitally and partly manually.

While online access to GP records is free, UK citizens can expect to [pay up to £50](#) for a hard copy of their medical files. And we haven't

even begun to talk about the additional charges for sick notes, copies of immunization records, and documents for insurance companies.

All of which might well get you wondering, “isn’t there a better way?” In a word, yes.

Why Healthcare Needs APIs

As it stands today, across the world there are woeful inefficiencies in even the most developed and effective healthcare services. Take this example experience:

1. Patient A reports to the doctor with fatigue. They are booked in for a blood test which they attend. They hear nothing back about it so call their local doctors’ surgery.
2. The secretary can tell them that their results are fine but nothing beyond that. Another blood test appointment is booked. They have to call for results again only to find that the test came back with no conclusive results.
3. Patient A reports to a specialist in the meantime who requires another blood test, only for Patient A to later find out that they are testing for the same thing as the first blood test they went in for.

Already you can see the problem here (aside from our patient running out of blood): people and/or systems aren’t communicating properly, which leads to wasted resources and wasted time.

It would be easy to write this off as “just how it goes,” but given that we already know that getting systems to talk to each other is something that APIs can do very well, we’re glad to see that they are finally being integrated into the healthcare space.

Using APIs for Diagnosis

[ApiMedic](#) is an interesting service that provides a symptom checker patients can use to get an idea of what they might be suffering from. Of course, there's nothing particularly new about that; everyone now uses a combination of WebMD and Google to find out what their symptoms mean.

What's interesting is the way in which ApiMedic is being used by hospitals around the world. Developers at Istanbul University Hospitals, for example, have embedded the system on their website to allow patients to book an appointment with the correct specialist.

While it's true that there is a risk associated with relying on a patient's initial judgement, it's something that will almost certainly be required as more pressure builds on the healthcare industry.

But it isn't just patients who can use APIs for diagnosis. To make quicker, easier and more definitive diagnoses, doctors need the most complete **EHR** (electronic health records) data that they can get their hands on.

[Doctorly](#) provides an open API for EHR that professes to deliver nearly real-time EHR data to apps built using the system. Doctors often complain about sluggish systems that, even when they do finally produce the requested information, give them incomplete data.

It appears that APIs could be a valuable tool for improving efficiency and coming up with diagnoses more quickly, for both patients and doctors, which could have important implications when a few minutes could mean the difference between life and death.

Improving Clarity for Patients

One of the biggest problems in the healthcare industry is communication between patients and doctors. Doctors often use terminology that's unfamiliar to patients and whether it's due to embarrassment or information overload, patients don't always stop to make sure that they know everything they need to.

[MedlinePlus Connect](#) is a service, developed by the National Institute of Health and showcased by the U.S. National Library of Medicine, that “converts” diagnosis, medication, and lab test codes into a simple text description.

Clearly this is useful for doctors as it means that, provided they have all their codes down pat, they can convert technical shorthand into plain English. However, there are also ways in which this could also be very useful for patients.

Rather than trying to memorize all test results, diagnoses, or the medication they need (for when they're trying to relay them to other doctors, chemists or even family members), they could tap a few codes into an app to get a detailed breakdown of the issues in language that they can understand.

Another indirect advantage of using APIs is that it potentially facilitates an easier and more intuitive interface through which patients can access their results. Jay Manciocchi, writing for Mashery, [suggests](#) the following use of healthcare APIs:

Although seasoned developers are used to APIs with long names that feature a challenging list of requirements, modern APIs tend to use web-friendly approaches, which are best known as RESTful (REpresentational State Transfer) and SOA (Service Orientated Architecture) architectural approaches. In healthcare settings, these APIs allow you to pull-up a web browser and type in a location like this to get some lab results:

http://ABCclinic.org/patients/John_Doe/labs/glucose_test/

If you've ever had to use an outdated web portal to access test results or make an appointment, you'll already know that they're rarely as simple as that hypothetical web address listed above.

Making Operations Easier for Health Professionals

As well as making things easier for patients, there are APIs out there that are changing the way doctors and other healthcare professionals run their business.

[Drchrono](#), for example, provides an [API and SDK](#) to help doctors manage their practice, bill their patients, and handle electronic health records in one place. And, even though we have no experience using it, it's worth pointing out that it looks a heck of a lot nicer than anything we've ever seen doctors using:

Source – [drchrono](#)

Because Drchrono places such emphasis on the importance of their **developer program**, they have a range of featured partners who integrate with their services. Among these are an electronic stethoscope attachment called [Eko](#), an app that fills cancellations automatically called [QueueDr](#) and a self-scheduling system called [NextPatient](#).

Until recently, a big issue has been that doctors have had to use multiple systems to manage their practice, make appointments, bill patients and so on. As a result, the process was fractured because it needed to be carried out on a number of different platforms.

Two things in play right now are solving that problem:

1. “One stop shop” services that are capable of handling several different elements of running/managing a practice.
2. More API-driven apps that integrate with a core system.

Previously a service might integrate with, say, a practice’s billing system but not its appointment functions. This doesn’t necessarily mean that the service is totally useless, but it does impose limitations on how effective it can be.

As services like Drchrono, which take a more holistic and programmable approach to running a doctor’s office begin to emerge, so does the possibility of integrating other services to make the whole process quicker and more effective.

The Trouble with Using APIs in a Healthcare Setting

In the UK, and elsewhere in the world, the healthcare services are publicly funded. While you might expect this to mean that efficiency and cost-effectiveness would be of paramount importance, the opposite is often true...as is often the case with [open data government initiatives](#).

Where public money is concerned, it’s usually a case of “if it ain’t broke, don’t fix it.” Even though many, like Kenneth Mandl and Isaac Kohane, [argue](#) that actually it *is* broken:

This myth continues to justify soaring IT costs, burdensome physician workloads, and stagnation in innovation—while doctors are becoming increasingly bound to documentation and communication products that are functionally decades behind those they use in their ‘civilian’ life.

Crucially, they also argue that “a few companies controlling much of the market remain entrenched in ‘legacy’ approaches, threatening other vendors’ viability.” However, things can’t remain the same forever, and we’re finally starting to see the healthcare industry begin to embrace new technology.

In their article on the untapped potential of APIs in the healthcare space, [Harvard Business Review](#) identified four main needs to accelerate their adoption:

1. Obvious financial incentives to encourage data exchange
2. Address privacy and security concerns, which is an ongoing concern for anyone transmitting data using the cloud
3. Development of industry standard APIs with transparent costs
4. Deal with issues relating to culture, such as the fear of losing control, patient-doctor relations, and workflow

Provided pilots and policies continue to be lined up to tackle these issues, there’s no reason why we won’t see even wider adoption of APIs and other technological advances to shake up the healthcare space.

Making Things Easier for Researchers

Writing for Mashery, Jay Manciocchi [points out](#) another benefit — this time an indirect one — that APIs in the healthcare space offer:

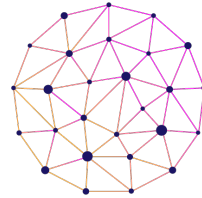
Modern APIs also offer the **standardization** of data. Each field must use consistent units and terminology, which can be challenging, given the information found in most healthcare records...Clinical decision support, public health, and research depend on quantifiable data, which APIs lead to.

If anything is going to speed up the adoption of APIs and the Internet of Things (IoT) in the healthcare space, this is it. Successful research leads to pilots, which leads to results, which leads to progress. Along the way, most of these activities also generate funding.

If we're really going to get cynical about it, the financial impact that using APIs in the healthcare space has on potential funding opportunities is one of the key reasons people and organizations will be more open to adopting them.

The Future of Healthcare and APIs

The most exciting thing about APIs in relation to healthcare is that they suggest a time, maybe not too far from our own, in which patient information is truly joined up.



Imagine an ambulance arriving on the scene of a car crash and being able to see their entire medical history. No more need to check for medic alert bracelets, organ donor cards, or evidence of pacemakers and allergies. Medics will instantly know that they need to be extra careful moving them because of a chronic back pain condition and that they can't give the patient any morphine because they reacted badly to it during a previous hospital stay.

But how, and where, is all of that data stored? Several experts have suggested blockchain technology as [a potential solution](#), with decentralized electronic health records available to all healthcare services that require it.

Healthcare expert Peter Nichol [suggests that](#) that each patient would be provided with a code and an address to unlock their data and enable access, with contributors to records being provided with another signature that combines with the patient's hash to authenticate their own access. All of which sounds just as, if not more, secure than the current system.

It's true that a world of more freely accessible healthcare data comes with plenty of worries of its own – such as how to make sure that confidential medical records never fall into the hands of employers, journalists or even unauthorized family members. Other risks arise with the involvement of healthcare startups that may not abide by proper protocols, as is the case with ongoing into Theranos, the [blood testing startup now under investigation](#).

Provided that the space can negate these concerns, making health-care more programmable will come with limitless possibility for progress, growth, and overall social welfare.

APIs Are Evolving The B2B Landscape



Phone, fax, and email. Archaic methods of communication are still the bane of corporate partnership exchanges. 700,000 fax machines were purchased between [2011 & 2012](#), and to this day, practitioners in healthcare and law adamantly believe fax to be the most “protected” form for sending documents.

Partners need to securely and reliably do business, and old processes can meet intransigence, even in the face of large tech momentum. Some of the traditional processes — like email — are here to stay. Nevertheless, partner integrations and information exchange is evolving, becoming more automated and programmable. Cloud computing is creating an explosion in the digital world, replacing

EDI ([Electronic Data Exchange](#)), phone, fax, and email with the Holy Grail for partner integrations — **RESTful API enabled B2B communication**.

But B2B processes are inherently slow to evolve — such complex systems can't possibly possess the [agility](#) of a startup to quickly adopt and test new methods. The fact is that thousands of connections and industry checkpoints are barring innovation in these complex corporate environments.

With these realities and forecasts in mind, what do we need in order to increase API awareness? How can we initiate industry momentum? The answer lies in properly **communicating a superior business value**. This means revisiting how APIs are **discussed** — with the goal to describe an inherently subject matter so that 'APIs' can rise above the constraints of D2D (Developer to Developer) communication, and enter the everyday B2B (Business to Business) dialogue.

In With The New, Out With (Some) Old

Just because old processes are ebbing doesn't mean they will fade forever. [Sumit Sharma](#) of Mulesoft predicts the future will see API communication replacing some EDI modes, increasing interoperability throughout the B2B world.

Though in the rear view mirror, phone, fax, email and EDI — the transfer of data from one computer system to another by standardized message formatting — are still huge and necessary players in international business.



The Clothing Chain Example

To hone in on a real world example, it's estimated that [Li & Fung](#), a global supply chain manager, processes orders for 80% of the clothing produced in China for the US. They handle the specifications for communicating with factories and distributors all throughout the world.

Within such an enormous system, coordinating order clothing production specs (various components like quantity, size, color shade, print, etc.) with thousands of factories throughout quasi-digital areas in rural China means spotty internet availability, stalling the advance of APIs to replace old modes of communication within that industry.

Sharma contends the reality is that EDI is here to stay for a while, but a hidden API behind all transactions would be ideal. Speaking with Sharma, a Li & Fung representative expressed interest in using [small cloud applications](#) to assist this process if it were possible.

B2B 15 Years Ago: The Progression

Let's take a quick jaunt through web service history to get some perspective on where we're at right now. 15 years ago, [service-oriented-architecture](#) (SOA) was the hot mode of B2B interoperability. For businesses facing partners, this was the first level of abstraction of their services. Then came the [SOAP](#) protocol for accessing web services, exposed as cloud services, and instigating machine-machine interoperability.

Depicting API history as the progression of lithic-wielding ape to RESTful human, Sharma outlines how monolithic apps led to architect-led APIs for SOA, leading to APIs for web, mobile, IT, and cloud. In the future, Big data business driven APIs represent the next step in this natural evolution. Sharma believes this trend shows

movement from a machine-machine process to a more accurate representation of the smart business-to-business process.

Why B2B Strategy Really Needs An API Strategy

APIs can increase agility by de-coupling and exposing business processes. Increasing machine to machine interoperability via APIs can erase human error, improve internal efficiency, and open up an organization to [new distribution channels](#). With the benefits from an [internal or partner API strategy so apparent](#), why aren't we seeing more APIs in B2B dealings? Why isn't [REST](#) more prevalent?

Sharma believes that it comes down to a lack of understanding, contending that "Application Programming Interface language" is inherently non-intuitive for business minded people. So, how do we convince business leaders to adopt an API strategy into their business curriculum?

"There is some paradigm shift that needs to happen so that we can get this into a business conversation"

The **cloud-computing** wave has taken off. With remote servers supplying thousands of businesses with remote processing, cloud computing has solidified itself as an important consideration for the modern business. Sharma believes that packaging an API strategy within the context of a "cloud-computing" discussion can help onboard some that are unsure about the technology.

We also need to convince business leaders why they need to deliver a new experience to their partners, with benefits clearly outlined. Functionality needs to be represented in relatable terms for those unfamiliar with tech specifics, and we must mitigate the risks involving API adoption and platformization. As B2B is built upon

relationships, API discussion needs to be transactionally designed in a way that fosters trust.

To this end, Sharma believes a solution lies in white-boarding:

- Using an **outside-in** thought perspective: what are we trying to achieve?
- Framing the API as a **product**: how can we describe services in a way that is in the business mode?
- **APX**: Increasing API **user experience** by designing first, before implementation.

To accomplish these goals, an API spec needs to satisfy the following:

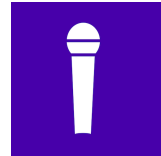
- Decrease **complexity** but retain comprehensiveness: Sharma encourages movement from WADL/ SOAP to **JSON based infrastructures**.
- Use lightweight systems with built in reusability
- Be **human readable** with clear parameter objectives and documentation.
- Reflect a structure that says: “this is what an API looks like”

Getting The Conversation Rolling

With the dawn of the first web APIs in 2004-2006, “what is an API” spiked in search rates. Now, we see a steady resurgence in queries, increasing in number each year — reflecting the API community growth and indicating an increase in general public knowledge and attention. Now is an ideal time to revisit how to refine our discussion on web APIs, and to address concerns — on security, longevity, control, and others — that deter **growth**. In working through these points, we can spread awareness and promote even more widespread adoption.

API Advocacy

Within the tech sector, evangelists spread ideas, promote developer programs, and are the face for a company. Sparking industry innovation requires a similar tenacity. It begins with [transforming individual businesses into platforms for growth](#) — a process that involves spreading awareness with evangelism and internal entrepreneurship.



In our eBook, [Developing The API Mindset](#), we outlined the benefits of introducing APIs into your organization. We also composed [email templates](#) used to disseminate knowledge throughout your company. Generating interest in this way helps introduce the topic and initiate discussion within an organization. When promoting an API strategy, it's important to refer to examples of API pioneers, and to follow [helpful resources](#) to keep updated on the API economy.

Convincing An Architect

Firing up API discussion often meets with intransigence from API architects. According to [Adam DuVander](#) of [Orchestrate](#), who recently gave a [presentation](#) at Nordic APIs in Seattle, some lead designers are squelching an API connected world.

These constituents might have 'architect' in their title, they may be the CTO, or even an opinionated engineer. Whatever their title, it's important to realize that certain team members will be resistant to change, associating API adoption with risks and certain stigmas. To help curb this, API evangelists should arm themselves with answers to these common pain points barring API adoption:

“We can't integrate APIs because of the lack of control”

An architect wants to touch a service. Accustomed to traditional in-house servers, an architect might be weary of cloud adoption. Functionally, the prospect of relinquishing control of third party data without granular access might not sit well either. Even with these considerations, the fact is that operations are accelerated *considerably* with API integrations.

Take for example the [Avalara tax APIs](#), which alleviate a common headache amongst eCommerce providers — sorting through confusing tax code. Surprisingly difficult to navigate, varying jurisdiction zones can segment tax to the block ([or even cut houses in half - as in the case of this peculiar Belgian town](#)).

Rather than maintain a log of thousands of varying tax rates, a developer can save time and resources by outsourcing this task to a provider that is an **specialist in that particular field**. This is the progression of leading internet startups, such as Uber, who [capitalized on the prevalence of numerous APIs](#) to construct much of their service. The mapping, location, driver optimization, payment, and rating services within the Uber app are all driven by cloud-based APIs, making it possible for Uber to focus on their core competency — finding drivers — and the user experience rather than become an expert in all fields.

Answer: The freeing of time and resources outweighs the lack of control, and will allow our business to increase its focus on our core business value.

”Can I change that software? Can I impact it? Can I touch it?”

Weary of potential downtime or latency issues, an architect desires close proximity to any data that is put into the API. Workarounds include allowing data downloads, or even offering an on premise managed option for the API. Having these options available can help onboard architects that are weary about that lack of fine

grained data control. [Factual](#) is an example of a company that does this — offering place and business listings provided via API with an option to download the entire data set for a fee.

Answer: Though we won't have local access, it likely won't affect processing speeds. If it becomes a problem we can always centralize the data and API.

"Is an API reliable? Can we count on consistent uptime?"

Another facet inhibiting API adoption in the enterprise context is the reliability and uptime of cloud-based services. Having a well-documented API is crucial for developer interest, but what happens if the actual service fails?

DuVander believes that we can avoid this stigma by properly communicating uptime. To ensure credibility, having a transparent log that displays API uptime is critical. Examples of this include the [Stripe system status](#) and live [twitter feed](#), which allow transparent and helpful status updates on downtime. Other models of quality developer-facing API status pages are [Facebook](#), [Twilio](#), and [Github](#).

Answer: Downtime is rare, but when it does happen it is communicated transparently. Additionally, support channels exist to help increase response time.

"How can we ensure the API service is secure?"

In the context of enterprise integration, security is of utmost importance. Though partner APIs are not exposed publicly, treat them as such with the proper security protocols. Securing systems with [modern approaches](#) that properly [delegate user credentials](#) and have a consistent policy for data access can ease the prospect of third party integration.

Answer (Hopefully): The API provider is well versed in security best practices, using an impressive [security stack](#) to properly distribute sensitive information.

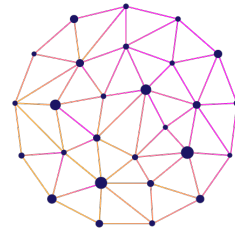
”What if the API service is deprecated? Dependencies will be left high and dry!”

Especially in the public API space, short lifecycles can make [product longevity a recurring issue](#). Google, for example, deprecates APIs nearly as often as they release new versions. So, how can partners avoid an unanticipated surprise? As the B2B context necessitates transactional agreements between both parties, contracts that guarantee consistent usage are more common, delineating access stipulations and longer operational phases. This is critical for banking institutions, for example, who require dependable technology that can withstand decades of use. As an API provider, you need to know [where you fit in the API business model chart](#) and clearly communicate your deprecation policies.

Answer: The provider/partner has a stable business model, and is willing to guarantee an operational lifetime that fits our needs.

Analysis: B2B World is Slowly Changing with APIs

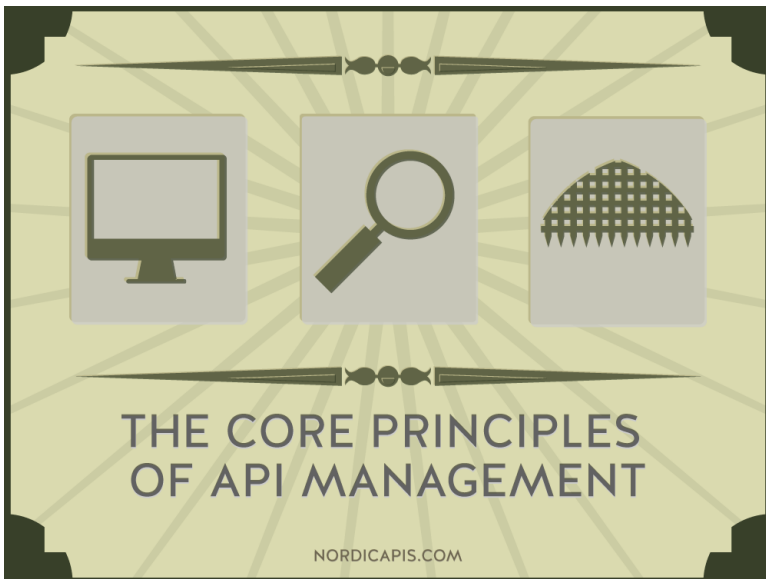
Adopt APIs into the enterprise context to accelerate business dealings — expediting a typical supply chain model, for example. With the dawn of cloud computing, companies are exchanging data and services at an ever growing rate. Collaborations are exciting, but can get messy if not executed correctly.



The consumerization of IT and B2B with packaged apps and applications is evolving, and the future will see more RESTful APIs driving this exchange of data in the form of [microservices](#) and fine grained processes that organically mimic true Business to Business and human processes. Remember that APIs have the transformative power to [reshape entire industries](#). To sum things up:

1. EDI is dying
2. [REST has eaten up most of the world](#)
3. B2B strategy is about relationships and human strategy
4. Evangelism can help initiate discussion
5. Tailor API discussion to a business context
6. Understand how to respond to roadblocks and pain points along the way

The Core Principles of API Management



API management is a term that has been used in the API economy for several years and has existed without acquiring an exact, universal definition. While most protagonists in the API community agree on the core themes of **API management** and why the subject is important, the specifics are frequently defined by the capabilities of commercial “API management” solutions. The term has become so synonymous with these solutions that more than one have it engraved in their brand.

The tendency to align the definition of API management to a vendor solution is exacerbating the fact that the management of APIs is not generally discussed with a great deal of objectivity, as it is skewed

towards the vendor's capabilities.

The dialectic on API management is therefore ripe for review. As the API economy continues to grow it is important that newcomers understand what API management is, why it is important, and how they should look to implement it. Past attempts to define the term have been done using surveys, cataloging the features of the most popular API management solutions available to organizations. Unfortunately, casting the net in this way obviously results in focusing on the *solution* rather than what API management should achieve: Looked at objectively, the features are an amalgam of other solutions and architectures, resembling SOA registries, Policy Enforcement Points (XACML), and so on.

Rather than representing the capabilities of a given toolset, API management is a generic practice that can be implemented regardless of any commercial solution an organization chooses to use. It is therefore important to arrive at a holistic and vendor neutral definition of what API management is.

A Definition of API Management

API management is the practice an organization implements to manage the APIs they expose. This is done either internally or externally to ensure that their APIs are consumable, secure, and [available](#) to consumers in conditions agreed upon in the APIs terms of use. Essential features API management should provide (rather than what specific solutions do) include the following:

- API management should provide a means for organizations to catalog their APIs, incorporating metadata such as the subject matter, description of the API (including different versions of the API that are currently available), human-friendly documentation, a taxonomy of the types of API

available, and runtime capabilities (such as maximum requests per second). The catalog should also register the state of a given API, including metadata such as the currently supported versions;

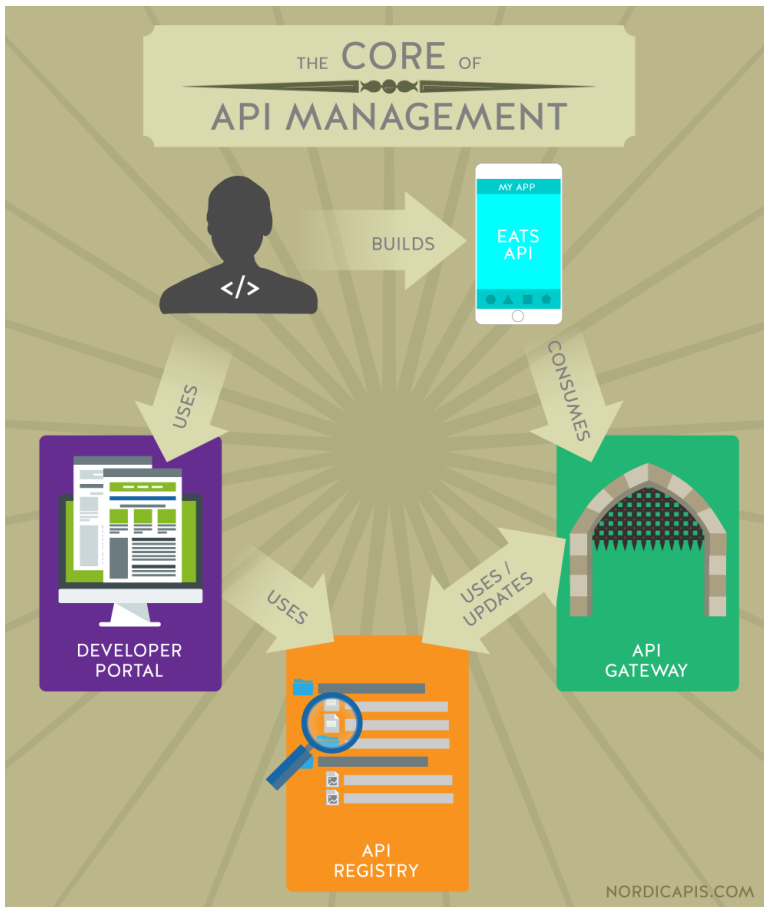
- API management should also provide a means to act on the catalog, exposing the APIs therein to internal and/or external developer communities with the ability to enforce security controls, consumption entitlements (in the form of mechanisms such as rate limits or quotas) and surface multiple versions as required. The distance between an API being catalogued and it being exposed as a consumable endpoint should be as short as possible, with the transition being equally seamless;
- An organization may expose APIs here that do not meet the organization's API "standards," or exist in a form that an organization does not want to expose to their consumers (as it closely coupled an external exposed API to an internal system that is sensitive to changes). API management should also provide the ability to transform the inputs and outputs accordingly, exposing a standardized form to the API consumers;
- Finally, API management should be the system of record for API utilization, embellishing the catalog with information regarding the actual runtime behaviour and characteristics of a given API in the form of metrics determined against key performance indicators. This information may include the number of API keys registered, average and peak requests per second, and so on: In fact, any data that is meaningful to the organization allows them to understand API utilization and plan accordingly for future enhancements or capacities. The information will also be used to help both monitor and monetize the APIs exposed, with the ability to make the data captured available to the organization's operational or billing systems as required.

There are clearly other capabilities that API management solutions in the marketplace offer; for example, the ability to expose SOAP-based services and databases as RESTful endpoints, perform orchestration to such services or directly monetize the organization's APIs by billing consumers using functionality implemented in their API management solution. While such features are clearly useful to many organizations who do not possess components in their architectures to perform these roles, we've deliberately kept them outside our core definition in order to center it on a set of features that *all* API providers need from their API management implementation. The key point is that many organizations will look to leverage components they already have in their architecture for these capabilities, and duplicating them purely for their API offering is likely to increase complexity, in terms of both technology and process.

Components

Now that we've established a more concrete definition of API management, we can look at the components that make up the solution. In terms of the [API management survey mentioned earlier](#), the components below are largely consistent with the majority of solutions found in the market at the time of writing and provide a convenient mechanism for grouping the features discussed in our definition. However, in terms of an architectural view, one should view these components logically rather than physically: All the components should be present in the logical architecture, but may be manifested in a different way depending on the API management solution.

In terms of an architecture, the components of API management are shown in the following diagram:



We'll discuss the role of each in turn.

API Registry

As we discussed in the features section above, the key to effective API management is having an inventory of an organization's APIs that allows API consumers to digest the characteristics of the APIs available, namely:

- **Features:** Describes what the API is designed to achieve in a form human beings can digest;
- **Structures:** A schematic description of the APIs, including URIs, data structures, security, etc.;
- **Capabilities:** What is the peak load the API can handle (both projected and actual) and the performance pinch points;
- **Sensitivities:** Does the API consume or expose any data that may be subject to regulatory or privacy constraint, such as payment card data, personally identifiable data, and so on.

The API registry provides this capability, holding data on behalf of the [API Gateway](#) and [Developer Portal](#) to provide a catalog of information for human beings to digest. The registry will also help an organization manage the [lifecycle of an API](#), cataloging the supported versions and their promotion or [retirement](#) from the organization's API inventory.

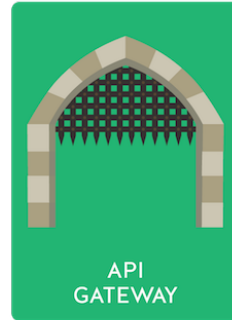


Going back to the point about logical vs. physical architectures, the API registry is the component that is likely to be manifested within other components in a physical architecture (most likely the Developer Portal), or indeed in many organizations may be federated across both API management and other systems in an organization's stack. However, an idealist view of the API Registry is that this is a discrete component, specifically implemented to harbour an organization's API knowledge.

API Gateway

If the API Registry is the brain of API management, the API Gateway is almost certainly the hands: It is where the business of API management happens as it is the component responsible for exposing the organization's APIs to its consumers (which can of course be internal or external).

In brief, the API Gateway covers the following key areas (for a detailed discussion of the capabilities of the API Gateway please refer to our previous [blog post](#) on the subject.):



- **Manifestation:** Exposes the organization's APIs to the outside world, acting like a proxy to route requests from external consumers to the API itself (as an aside, many API management solutions can also support backendless APIs, implementing the entirety of the API on the Gateway itself);
- **Security:** Acts as the gatekeeper to the API, applying [security mechanisms](#) on behalf of the API. Applying security at this outer perimeter is consistent with the cyber security principle of "defensive in depth" and is an important feature of the API Gateway's capabilities;
- **Entitlement:** Allows access to APIs at the agreed upon rates and either limiting or managing traffic;
- **Standardization:** Presents a uniform suite of APIs to consumers (according to any API standards an organization might have);
- **Logging and data capture:** Captures the information required to understand API utilization, for a variety of purposes discussed below.

In order to expose, secure, and manage an organization's APIs

the API Gateway clearly needs to work in collaboration with the API Registry, ingesting information about the APIs and how they should be exposed. The API Gateway should also establish a **feedback loop**, supplying the API Registry with management-level statistics on API utilization, in order to help maintain an accurate picture of how the organizations APIs are used and thus their relative importance for future capacity and investment decisions. Finally, data captured by the API Gateway should be available for ingestion by any other system that requires it, for purposes such as monitoring, [monetization](#), [analytics](#) or insight.

Developer Portal

The final component that forms our API management architecture is the Developer Portal. If the API Registry is the brain and the API Gateway is hands, the Developer Portal is the ears and mouth, where an API provider both listens and talks to the [developer community](#). The Developer Portal provides the human interface to an organization's APIs, providing a quality user experience (whether internal or external) and helpful tools and resources for building applications that consume the API. Moreover, the Portal provides the facilities for developers to manage their engagement with the organization.



To provide an effective management experience the Developer Portal should include the following features:

- **Learning:** Provides the facilities (documentation, how-tos, API descriptions, etc.) that a developer requires to understand

the APIs on offer;

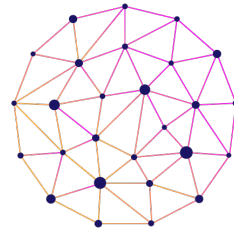
- **Exploration:** Delivers mechanisms that allow the developer to experiment with an organization's API before they even write a stitch of code — interactive explorers, sample code generators, [virtualization](#), and so on;
- **Implementation:** Helps the developer manage how their application(s) interacts with the API by providing the ability to register for a given price plan (if the API implements a charging model), create [API keys](#) and secrets, register callback addresses for OAuth, register webhooks, etc. (the specific features being dependent on the features of a given organization's APIs);
- **Feedback:** Allowing developers to understand the metrics captured by the API gateway in the form of dashboards and insights. This can also be useful for internal stakeholders who will want to understand how the organization's APIs are be utilized, for the purposes of operational monitoring or monetization (although the organization may choose to provide this information through existing tools rather than API management of course).

The Developer Portal therefore plays an important role in the overall [developer experience](#), working in collaboration with both the API Registry and API Gateway to provide the information that developers need to correctly understand the organization's APIs. It will also present usage data that helps API consumers understand how they are using the APIs, which can be important for improving their own applications or ensuring they are getting value from API consumption.

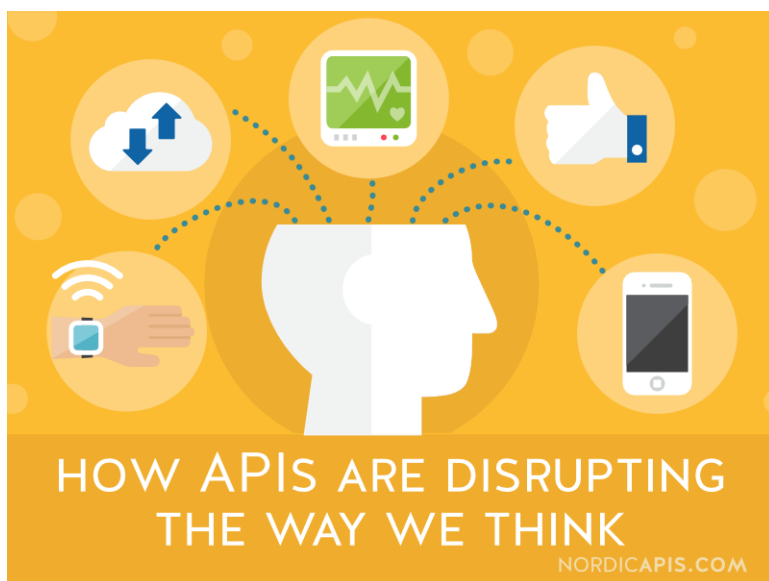
Analysis: An Attempt at an Objective, General Definition

In this chapter we've attempted to set out a vendor neutral interpretation of API management. Clearly retrofitting a generalized framework for comprehending API management onto all the interpretations of the subject offered by solution vendors is going to be difficult.

However, establishing the **core principles of API management** and what API management solutions should provide will hopefully help organizations who are implementing a practice of API management to apply objectivity to their view of what their requirements and needs really are, rather than simply adopting trends or feature-bloating on third party API management solutions.



How APIs Are Disrupting The Way We Think



The concept of *disruption* - or hustling - has been given regal status across businesses, startups, and tech circles in recent years. With such great emphasis placed on change, user experiences are inevitably facing evolution as well. **Application programming interfaces** or **APIs** have great transformative powers to disrupt business, but are they also altering the way we think?

In a word: Yes. The API is actually the driving force behind most of the **digital disruption** in the consumer space happening right now — cloud-based infrastructure, mobile apps, Facebook logins, online shopping and viewing — it's just that most laymen don't realize it. Indeed, APIs are surreptitiously increasing connectivity and

enabling unprecedented services, disrupting the way we interact with the world.

In this piece, we endeavor to talk about how APIs are not only disrupting most if not all traditional business models, but are changing basic assumptions of how we interact with the world... and just how developers can adapt to join in the world-changing fun.

Disruption or Destruction?

There's a fine line between disruption and destruction, especially depending on which side of the tech you are. Cofounder of the [Dopter open data and API consulting firm Andreas Krohn](#) discussed with Nordic APIs how APIs are disrupting even the previously analog world. Krohn offered countless business models disrupted — or perhaps destroyed? — because of the addition of an API backbone:

“The hotel industry thought it was previously disruption-proof, until AirBnB came waltzing online to empower consumer choice and to allow anyone to become hotel management.”

And we all know what happened to Blockbuster, which a decade ago boasted 9,000 stores and 60,000 employees. “They were seen as the dominant player, you could never move these guys,” Krohn said, but now Blockbuster is a distant memory, pushed into the shadows by Netflix, who are now responsible for more than [34 percent of all Internet traffic](#) during peak U.S. usage time.

This enormous shift has been made possible by Netflix's internal API, which handles “two billion requests a day,” according to Krohn. Their internal API enables Netflix to rapidly churn out new

apps to expand to new platforms — smartphones, X-box, smart TVs, and many other devices— dramatically cutting development time as only small components are necessary for each device. “They don’t do the whole system. They can easily package their data from the API.”

Through its use of APIs, Netflix has disrupted the entire television industry. Krohn says, while open and public APIs are the wave of the future, Netflix is an example of the immense value that’s generated from an internal API, even when done simply between departments.

Of course, when you mention interdepartmental APIs, Amazon’s “eat your own API” naturally comes to mind. For years Amazon has required that all data-based communication between departments be done via API, naturally positioning Amazon to lead disruption in a world where APIs are becoming more and more ubiquitous.

Amazon’s disruption began with the decimation of brick-and-mortar bookstores — shutting the doors of Barnes&Nobles and Borders— but they didn’t stop there. They found a potentially larger market in going after technical pioneers IBM, HP, and Dell — some of the world’s largest creators of hardware servers. “Amazon today provides the same services as this hardware did, via APIs. If you want a new server today, don’t go out and buy one. You can create one very quickly online and run it on Amazon’s clouds,” Krohn said.

“What started as a bookstore disrupted the hardware industry because they took what they built themselves and built a product out of it. They needed to have a website that was up all the time, people can buy books any time of the day, any time of the night—that’s a lot of infrastructure.” Amazon then looked for the core of the business and discovered its the IT and infrastructure,” thinking, “We can sell that too.”

The [value of Amazon Web Services](#) is now \$50 billion, giving obvious credence to finding your core value.

Krohn reminds that “It’s important to be aware that every industry can be disrupted. Your industry, whatever it is, is either slowly being disrupted right now or will be soon. That’s just the fact of life.”

Mark O’Neil, VP of Innovation at [Axway](#), sees the case of Dun and Bradstreet (D&B) as [another superb example of APIs disrupting traditional business](#). A long established credit approval company — in fact, [Abraham Lincoln](#) worked for them — DNB has innovated [with their API](#), enabling D&B lookups to be performed from within third-party apps, or within SaaS services such as Salesforce.com.

“This is a great example of a traditional business which has successfully leveraged APIs, creating a new revenue channel and disrupting the industry.”

O’Neil, an expert on API management and a proponent of [API-first](#) strategies, also cites [First Utility](#), as having destructive potential to alter the electric utility industry within the UK. They help users easily switch utility providers, aided by an API that enables customers to receive quotes and sign up for their service.

“In this way, their API is disrupting a whole industry”

The Biggest Disruption of All: APIs Change Our Expectations

[Social networking theorist Brian Solis](#) once wrote that “technology evolves faster than our ability to adapt.” But while us humans are inherently resistant to change, we are adapting faster and faster to technological trends. Krohn compared the number of decades it took society to adapt to the telephone to the number of years it’s taken to adopt smartphones as a way of life.

Within this escalating landscape, “APIs make it quite possible for us to adapt to new requirements,” Krohn said. This disruption, happening quicker and quicker, is often created by two components: the technological advantage of a new player, or the advantage of a disruptive business model.

But, getting to that core, how exactly do APIs disrupt? They affect businesses and users by subtly altering our **expectations**. Ten or even five years ago, we wouldn’t have been asking “Why isn’t there an app for that?” “Why can’t I just do it on my phone?” “[Why can’t I log into this site simply via Facebook?](#)” “Why can’t we quickly integrate our systems?”

There’s no doubt that consumers are dramatically more technologically demanding, and it’s all the API’s fault for changing our assumptions, whether we realize it or not.

How Exactly Are APIs Disruptive?

APIs enable businesses to hit the ground running. Here are some reasons how:

- **Speed to market:** You want to do it now. APIs facilitate consumer expectations for rapid fulfillment.
- **Leverage:** With APIs, a small startup can do as much as a large company, and sometimes even more.
- **Silos are Ripped Down:** Data is mixed and remixed everywhere.
- **Productize & Profit From Your Core:** Businesses can deliver their core assets or services far more easily. The API is a portal, but data delivery is really at your core.

APIs Are the Digital Glue Holding Our World Together

One could argue that all of the important tech trends outlined in [Gartner's 2014 research](#), (except for perhaps 3D printing) come with an API backbone. Without APIs, more than “half of the major technological trends could not be possible. **That is important. That tells you something,**” Krohn said. “How basic this technology is. It’s kind of an infrastructure.” If you expect to efficiently access information from anywhere and to share data easily without the capital expenses costs, you have to reach for the sky — and APIs are the stairway to the cloud.

APIs are everywhere, whether or not people know or expect it. As technology advances, we demand a seamlessly integrated experience from one device, app, or function to the next. While the average consumer is accustomed to having technology run in the background, they are happily ignorant to the API architecture that’s behind it all.

“The greatest example is Facebook. They would not be where they are today without an API. Every time you see a login with a Facebook ID on a website, they’re using an API. Every time there’s a Share this Web button, there’s an API. Every time you use an application that somehow imports your friends...they’re using APIs,” Krohn said. Social networks are “using a lot of people and APIs to market their content and their presence because Facebook wants to own your online identity.” To own identity online means you must have the ability to be omnipresent.

Forbes has dubbed [APIs as the “digital glue”](#) that’s holding our personal and professional worlds together. But while the average consumer may be blind to this adhesion, certainly the business world is taking notice. In fact, Gartner says that last year about [three-quarters of Fortune-1000 companies](#) were using public APIs to grow their business.

How Can Today's Developers Take Advantage of API Disruption?

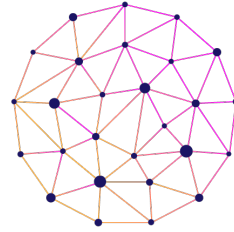
Now that big business is finally recognizing the true value in the API economy, it's time for developers to grab a piece of that pie. Krohn offers tricks to help developers and API practitioners end up on the winner's side of API disruption — and not left behind on the side of destruction.

1. **Educate the Masses:** “It’s important that this is not an IT issue. Marketing needs to know about this. CEOs need to know about this. Sales need to know about this. What’s an API and how can it impact us?” If an API product is IT only, it’s bound to fail because without these cross-organizational connections, budgets won’t be approved and in-house developers will move on to other things.
2. **Publish:** This is one of the best ways to learn the process of what’s going on in the API space. “Start to learn what’s required; start to learn to educate your organization about it,” Krohn said, driving home that your writing must be based on external requirements. Constantly ask yourself: “What would somebody from outside our organization actually need?”
3. **Talk about APIs:** As part of an agile, continual improvement process, consistently talk to the people using the data. Spread your enthusiasm!
4. **Use APIs:** Of course, eat your own dog food, but experiment with other APIs that could add benefit to your organization. This helps you stay up to date, scope the competition, and realize the holes in the API ecosystem that perhaps you could fill.
5. **Experiment & Play:** Host [internal hackathons and exploration days](#) that encourage open innovation to combine internal data processes with external ones. “It’s useful and it’s

fun too. And it's a good way to get different parts of an organization to talk to each other," said Krohn.

Analysis: Drive API Disruption

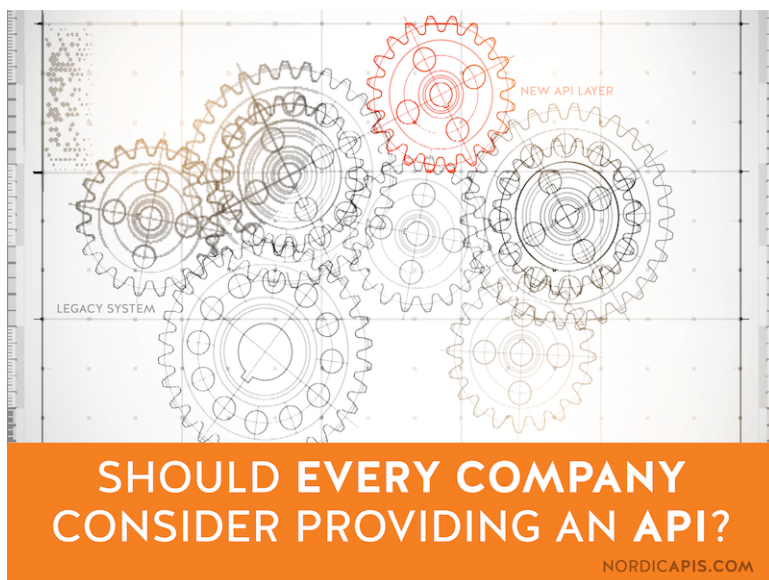
APIs are the digital glue holding our world together. They are pulling together our technological world and now they're removing barriers within the business world, enabling partnerships that were never possible before. Developers now have an infinite amount of power in deciding if they will be disrupted or lead the disruption.



Let's review how to drive API disruption:

- Look at what you really do. Examine the business model and infrastructure at your core. Are there valuable and extendable use cases for your data or services? Explore them.
- Be an API evangelist to the masses. APIs are still rather an unknown concept. It's our job to teach the business value proposition.
- Focus on your specific industry and study the API landscape. What holes could your business fill within the API economy?
- Consider profitable partnerships. As Krohn said, "if you don't have the ability to partner with other companies, your competition does. And then they turn to the competition instead of you."

Should Every Company Consider Providing an API?



For many companies, providing an API is seen as an IT matter exclusive to internet giants like Twitter, Facebook and Google, startups like [Algolia](#), [Wit.ai](#) and [Context.io](#), or for government agencies to open data to the public. But are APIs really limited in that way? In this article we denounce common myths associated with APIs and encourage others to join the digital revolution.

An API provider is an organization that exposes data and/or capabilities through a programmatically consumable service or an Application Programming Interface (API). [Nordic APIs](#), [API Platform Defined: When](#)

an API Provider is a Platform

Common Myths Surrounding APIs

When some people read ‘providing APIs’ they picture only the visible part of the iceberg, assuming the three following falsities:

- Providing APIs means providing *public* APIs
- Providing APIs is an *IT-only* matter.
- Providing APIs is for **internet startups** and **giant corporations**.

Myth #1: All APIs Are Public

Having a **public API** offering is not the only method of *providing APIs*. Regardless of technical details, from an organization’s point of view there are actually three types, each with unique benefits:

- Private API: Consume the API for internal use
- Partner API: Provide the API to selected partners
- Public API: Offer the API to anybody

Loosening restriction in that order is [now a common API strategy](#), but each type is valuable in itself and can bring unique benefits to a company.

Myth #2: Providing APIs Is an IT-Only Matter

Since ‘API’ stands for ‘Application *Programming Interface*’, [many people are put off by the technical overtones](#) and assume that it’s

an *IT-only* matter. That's absolutely not true. APIs have the power to transform your company far beyond the IT department by:

- Transversally impacting how the organization does business.
- Enabling your company's digital transformation to be more customer focused.
- Hiding information system and organization complexities.
- Acting as a new product for the business.

As their impact can be huge, reducing APIs to an IT-only matter could be a terrible mistake for a company. Therefore, they should be designed and handled by both IT *and* business.

Myth #3: Providing APIs Is Only For Internet Startups And Giants

An organization does not need to be a startup, internet giant, or government agency to be concerned with APIs. There are many other types of companies with great API programs, such as:

Bechtel, a construction and engineering firm

[Bechtel Corp.](#), the largest construction and engineering firm in the United States, engages an API strategy to facilitate access to its monolithic system. Their strategy allows the company to accelerate productivity, enable [mobile access](#) with up-to-date relevant data, and diminish paper-based management in the field.

[Source](#)

Dun & Bradstreet, a business information company

[Dun & Bradstreet, Inc.](#) is an American public company that provides commercial data to businesses on credit history, business-to-business sales and marketing, counterparty risk exposure, supply chain management, lead scoring, and social identity matching. Using APIs, D&B provides data easily and directly into their customer systems, applications, CRM, or ERP — regardless of technologies used.

[Source](#)

Marvel Entertainment, a character based entertainment company

The [Marvel Entertainment](#) public API allows access to data on over 30,000 comics and 7,000 series. Everything from cover art, character info, and comic book crossover events are accessible via the API.

[Source](#)

Absolut Vodka, a spirits company

[Absolut Vodka](#) provides the [ADDb](#): Absolut Drink Database. This is an API for Absolut's drink recipes and other related assets that reduce their advertising campaigns' time to market. [Source](#)

Your company?

For more than forty years I have been speaking prose without knowing anything about it.

- *Mr Jourdain*, [The Middleclass Gentleman](#), Molière

Like Mr Jourdain, perhaps your company is already an API provider without knowing it. For example, if your company has a mobile

application or a website plugged into your information system (even if it's for an internal purpose), you may be in that case. If your company exchanges data with partners via web services, you may be in that case too. These *APIs* may not correspond to the current standards but it may be a good idea to think about capitalizing on them.

Developing an Internal API: A First Crucial Step With Many Benefits

Learning how to walk with in-house APIs is the *must do* step in [every company's API journey](#). Secured within the walls of your company, you can easily learn from your mistakes. If your company does not foresee, *for now*, the necessity of exposing data and/or capabilities to *others* via APIs, being *your own* API provider can already have many benefits.

From The IT Department Perspective

Though we're arguing that APIs should not be an IT-only matter, they are still a *Programming Interface*, so it's fairly common that IT will introduce them to the company and be the first to benefit.

Increasing Agility

The best way to introduce APIs into your information system is to add a new layer on top of your legacy system. The legacy layer can continue to live as before with its own (maybe longer) life cycle.

The new layer will have to be handled with a *startup mindset*. It must have a faster and more volatile life cycle ensured by a lighter governance and the use of (probably) new and different technologies. This new layer is a *logical* one — you're under no

obligation to have it within your actual infrastructure or to use only a single technology to build it.

Of course, this new layer will have to compose with the legacy one, but it will be easier to create APIs by orchestrating and adapting what comes from the legacy layer than modifying the legacy layer. New projects could also be totally handled within a new layer if there's no need for plugging to the legacy IT. This new layer may allow you to integrate in a more simpler way other APIs (Google Maps, Facebook, Twitter, USPS, Best Buy...).

With this two level information system, your IT is ready to respond to any new need with increased agility, whether it be a new mobile application, desktop application, web site or web service on top of your existing information system.

Creating Solutions With Building Blocks

This new layer does not only allow you to develop with more agility, it can also allow you to develop with more efficiency. By using an [API first approach](#) you can build reusable API building blocks instead of specific software solutions that will only respond to a specific need. Without having to recode everything, your IT can respond to any new need with more efficiency, inducing a faster time to market and decreased cost.

Refreshing Your Legacy IT

This new layer in your information system may also help you introduce new technologies and new ways of thinking that can improve your legacy IT. The new API layer comes with new (or not so new) ideas concerning [documentation](#), [design](#), development, and [life cycle](#) that will disseminate by capillarity to your legacy IT.

Not everything can be handled by separating the new API layer and the existing legacy system. Eventually, you will have to [evolve](#)

your legacy IT. It may be the moment to evolve towards the new technologies introduced by the new layer and perform a complete refactoring of some parts instead of ongoing palliative care on old unmaintainable systems. Being a central node of your IS, this new API layer could also be cheap way to start a big data program as it will deal with a lot of your data.

A Better Environment for IT Employees

Working on a paralyzed information system that doesn't evolve easily and prevents the introduction of new technologies can discourage many developers, architects and anyone involved in maintaining this old machinery. This increases employee turnover rate and makes hiring new IT people very difficult, as nobody will want to diminish their professional value on such an old system. By introducing new technologies and new challenges, APIs can make your company a better place to work in.

From the Company Perspective

For the company, an *API improved IT* can now propose a *light IT*, bringing new, once undreamed of, possibilities for the whole company. Here we will address five ways that APIs can *illuminate* an organization, bringing many benefits to internal operations.

1: Unleashing creation and innovation

light

\'lit\

adjective: capable of moving swiftly or nimbly.

adjective: requiring little effort.

noun: spiritual illumination.

By enabling a *light IT*, APIs can reconcile IT with the whole company and create new momentum. IT and business should work together to build the APIs that will be the backbone of the company. Capitalize on the IT improvement and deliver or evolve products easier, faster and for less cost.

2: Putting “Shadow IT” under control

light

\ˈlɪt\

noun: the form of energy that makes it possible to see things.

With a paralysed information system inducing high development costs and long time to market, some employees tend to try to find other solutions on their own...

Shadow IT is a term often used to describe IT systems and IT solutions built and used inside organizations without explicit organizational approval. It is also used, along with the term “Stealth IT,” to describe solutions specified and deployed by departments other than the IT department.

[*Shadow IT definition, Wikipedia*](#)

These solutions tend to be hard to maintain for non IT departments (which often pass the hot potato to IT in the end). As they aren’t officially sanctioned, these solutions often do not meet the organizational requirements concerning [security](#) or reliability, introducing risks to the company (costs, loss of revenue, reputational risk, and others).

APIs help avoid the use of shadow IT or at least propose a *controlled* version of shadow IT. Once APIs have been defined (by IT *and* non IT), it’s more feasible to allow non IT folks to build whatever they want (within certain boundaries) using these APIs.

3: Offering new ways of working

light

\'lit\

a medium (as a window) through which light is admitted

As APIs enable new ways of accessing information system, this means new ways of working. With APIs, it becomes easier to provide system access from outside the confines of the company.

For example, field-based personnel can access real time data while being outside the company's network on smartphones. You can give tablets to your representatives and allow them to subscribe to products directly from a customer's house.

With APIs it's simpler to integrate your data with existing solutions. No more costly development to plug a new software solution to your information system. We can even imagine pushing the envelope and having an IFTTT or Zappier equivalent for internal APIs, giving company internal users the same customization of daily work processes as they have on their personal smartphone.

4: Enabling disposable products

light

\'lit\

adjective: casual, occasional.

By reducing cost, development complexity, and by introducing reusable building blocks, your company can now even build ephemeral products, such as websites or mobile apps, that only need to last the length of a marketing campaign.

5: Providing knowledge of internal data and statistics

light

\ˈlɪt\

noun: something that enlightens or informs.

To build APIs that hide information system and organizational complexities, you need to analyze your system in order to restructure or rebuild lost knowledge. Therefore, APIs will give you a simplified and comprehensible vision of your internal mess.

If the API is the backbone of your company, consumption statistics can create a knowledge of how your data and processes are being used. Better understanding your data processes can create a great base for further innovation.

Providing APIs to Others: The Next Step

After learning to walk as your own API provider, your company may wonder if it's worth offering its API to others. Many of the benefits earned by being an internal API provider are expanded by providing APIs to others, but entirely new benefits also arise with this opening.

Preventing Data Hold Up

Many companies realize the risk that somebody *will* open their data sooner or later, whether they provide APIs or not. Personal finance management is a good example of *data hold up*.

Banks have been sort of robbed by account aggregators. These account aggregators like Yoddle, Intuit or FiduCEO started by scraping data from banks websites with agents logging in on a customer's behalf to offer new services to these bank customers.

Providing APIs to others to control is a serious option for many companies. It would be a terrible blow to let others harvest a company's fruits without reaping any benefits from this exchange.

Facilitating Working With Partners or Within Large Groups

Just as APIs can facilitate integration with software used within your company, they can also easily facilitate data exchange with existing partners — as exemplified by Dun & Bradstreet's commercial data exchange processes. If your company is a large group composed of multiple entities, APIs can also facilitate cooperation between entities, especially if you apply [Amazon's rules for success](#).

Enabling New Business Opportunities

By having off-the-rack APIs instead of tailor-made software solutions, you can easily propose your service to new partners or even to the public. These new opportunities will depend greatly on your sector and how you market and open your APIs. Possibilities are endless, but some examples include:

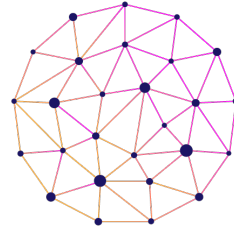
- Improving public reputation by offering data transparency on your product composition or your financial data.
- Letting other people innovate to build something that your company doesn't have the resources or time to accomplish on its own.
- Monetizing access to your API.
- Extending your customer base to small companies or developers who can't afford the cost of developing such services on their own.
- Increasing your data volume by offering free APIs.
- ...and more

Analysis: APIs *Must* Be a Concern For Every Company

What other conclusion could we have?

Providing APIs must be a concern for every company. Whether we focus on internal operations, partner integrations, or public access, APIs come with a myriad of benefits to a business with an information system, and also benefit everyone within the company. Join the

revolution — don't be on the wrong side of industry disruption.



The Future of the API Space



[Forrester](#) calls the **API** the digital glue holding our world together. [Forbes](#) says it's the key to unlocking the **digital economy**, while [TechCrunch](#) predicts it'll unlock the **data economy**. And, here at Nordic APIs, we've told you how the API is the driving force behind **smart cities**, the **Internet of Things**, and more.

"APIs are touching pretty much every part of your lives, how you get ahead in business, getting data. You're going to need APIs," [API Evangelist Kin Lane](#) told Nordic APIs. Whether we realize it or not, Lane argues that [APIs have completely disrupted the consumer psyche](#). He says the API is teaching us lessons about how to be social, offering examples starting back with Flickr and Del.icio.us and moving into the more mainstream oversharing of our lives

on Facebook and Twitter, and to how APIs have enabled us to professionally collaborate from anywhere.

Lane takes it a step further by saying that the API is the enabler that has allowed us to move our lives fully online, including, “our banking, photo storage, our music. All of this is enabled by APIs and the cloud. Our finances, how we trade stock, manage giftcards, payments with currencies, social, cloud commerce.”

It seems that the business world has finally realized what those in the back end have known for awhile — the **application programming interface** is no longer the future of our connected world, it’s the current reality. In fact, APIs’ growing popularity is even being analogized to [what websites were back in ‘97](#): most companies use them and almost the whole public is interacting with them, without quite understanding what it is or its value yet.

Now, we need to focus on the next direction on where APIs are headed. That’s why we talked to some of the biggest names in the API world to learn what they foresee is the **Future of the API Space**.

API Indicator #1: API Becomes an Accepted Tool to Grow Your Business

“APIs allow businesses to liberate their data, and monetize it,” VP of Innovation at [Axway software](#) Mark O’Neill told Nordic APIs.

When it comes down to it, an API isn’t the magic recipe for growth and not every business needs an API. “Can we pay for this?” has become the most important question when debating whether or not to develop. That being said, more and more businesses are finding ways to use the API as a fast solution to internal and customer needs.

“The future of APIs is that every call you make is driving transactional value for the provider of that API,” explained **John Shee-**

han, founder of [Runscope API performance monitoring tool](#), and previously of API-driven apps Twillio and IFTTT. “People should focus more on [building business value](#) than focus on building the API value. Find the underlying business value and how you can augment your API for it, not try to find a business value for your API.”

John Musser, founder of [API Science] (<https://www.apiscience.com/>) and [ProgrammableWeb](#), offered the questions to guide businesses in API decision making:

- How do I make money from this?
- Who is it for? For me? My partners?
- If I open an API, what is the core value I am offering?
- Is it [secure](#)?
- How do I use it myself, in our own apps and services?
- How do I justify the cost and effort? Are there multiple ways I can get ROI on this?

For Musser, the future of the API for developers is simple: “To save money, make money, be more agile, to disrupt a new industry.” He says that [Twilio Integration as a Service \(IaaS\)](#) is really the best example. “These companies are [API-first](#). These whole businesses are based on providing services to developers.”

He continued that “businesses need to look at APIs not as APIs, but as a *conduit*. Why am I doing it and who is it for? Am I doing it to make money? To save money? For broader reach? Partnerships?”

O’Neill shared some great examples of traditional businesses expanding via APIs, including Abe Lincoln’s former employer [Dun and Bradstreet](#) who created a new revenue channel by offering their data to Software as a Service (SaaS) like Salesforce, as well as newcomer electric company First Utility. “They enable customers to easily switch from incumbents to First Utility, though switching websites like [USSwitch.com](#) which call their API to receive quotes and sign up for service,” disrupting a 100-year-old industry.

API Indicator #2: Microservices Architecture Allows All Departments to Scale

We've already covered growing trends for [designing APIs for the developer experience](#), but we need to remember that integrators and developers aren't the only users of the API. There's a broadening trend that backroom developers are actually focusing more and more on how to solve the needs of other departments more quickly.

The increasing popularity of [microservices architecture](#) is allowing developers to become more [agile](#) in their software update releases, as building smaller, more contained features allows for shorter development cycles and shorter distance to market.

Through trends like microservices, APIs are becoming the final puzzle piece to building organizational-wide digital strategy, as the API comes out of the IT department and into the board room.

For example, APIs will continue to enable marketing and market research firms, and among other non-IT business areas, to use APIs to create short-term solutions or **situational applications** that serve as answers to short-lived problems.

API software consultant and founder of [LaunchAny](#) **James Higinbotham** calls this the concept of short-term application development:

“Businesses often times have the need for a very short-term application to be built, primarily around the marketing space. Marketers often run very short burst campaigns that require a micro-website or a micro-application. If businesses realize that APIs are not necessarily a technology solution but it's a way to implement **componentization of organizations**, then the business side can use these APIs to construct these

short-lived applications to achieve what they are trying to achieve — marketing, internal reporting, project timelines and [other] requirements.”

This all fits into a broader necessity for APIs being created to enable businesses to respond rapidly to changing customer needs, as well as internal ones. Higginbotham even says you may not even need to sell the term “API” to that marketing team, but rather sell a faster solution to their needs. “IT often runs on a longer cycle, even in the agile world,” he said, which leads to a clash with campaign-driven marketing departments. With APIs, “the marketing department doesn’t have to work with IT, they actually become the part of IT.”

But this movement away from the monolith structure isn’t just going to help outside the IT department, but help software scale as well, allowing different parts of systems to work independently, reducing time to market and to resolve errors.

Sheehan echoes Higginbotham:

“I think the future of APIs is a more concerted effort of companies of all sizes to really build out the SOA [service-oriented] architecture we dreamed of ten years ago but didn’t exist. I think that managers in the long run will find the flexibility to make them more responsive to business needs. [At Runscope] we have lots and lots of small services. It really has allowed us to focus on new features and scaling...allows us to kill off features, to rewrite entire components, without letting it affect the entire system.”

O’Neill continues the sentiment with “In your opinion, what should be the priority focuses for excelling the API space? APIs have gone from being one-off projects, to now being core to the organization. It’s important that APIs are linked to the rest of enterprise architecture. They should not be air-gapped away,” he said. “Excelling

in the API space means ensuring that your API is not a “science project”, but instead is part of enterprise infrastructure, from Day 1.”

When it comes down to it, it’s all about finding the way to make changes faster and to respond to more business needs because “the business changes faster than we could ever develop our software.”

API Indicator #3: Are Public APIs Losing Their Luster?

The wave of Public APIs is ebbing. What was once a hot trend doesn’t make sense for a lot of companies and can actually risk your good name if done too hastily.

While most of us are as enamored with the idea of the **Public API** as we are with open source, as Sheehan puts it, “we always thought that the most interesting API data wasn’t happening in the public or the API economy.” When it comes down to it, internal APIs may inevitably carry more valuable data.

Musser rightfully calls Netflix the poster child for APIs. Indeed, the cable company’s biggest threat has an entirely internal [API-based business model](#), “but they’re not trying to go to hackathons to have every developer use it.”

Does a company need a Public API? If you’re a Twilio or Clarify, where your API is your business, then absolutely. But otherwise it must be remembered that an API is one feature among your overall product offering and thus, as with all features, a cost-benefit analysis must be made.

As Musser puts it, “Open API is one flavor, one strategy, but it’s by no means the only or necessarily the best strategy. It depends who you are, what you’re trying to do, and it’s a mistake to sort of jump to that conclusion” of an opening an API too early.

It doesn't mean your API can't go public one day, it's just that it's a risky move on the first day. "Think of APIs as an evolution. I think it's quite natural to start off as having an API as an internal API and later you open it up to partners, and then, after that, a broader set of partners, and then later anyone," Musser mused:

"You don't have to boil the ocean from day one."

To hum the same refrain, the secret to API success is knowing *why* you are creating it. Ask yourself: What's its purpose? Why would it need to be public?"

Higginbotham agreed with Musser by saying, "I'm seeing more organizations desiring to build internal APIs, particularly larger organizations that are using Private APIs because they are still trying to understand how much they want to release to the public."

However, he contends that "the best of programmers are modeling their APIs after Public APIs, in case they want to release it." That means that [security is a priority from the start](#), which of course is a wise practice, whether it ever goes public or not. "I think over time, we're going to see more of these Private APIs beginning to emerge" to the public, Higginbotham said, but that they will be coming from APIs that were internally purposed but externally minded from birth.

And while public APIs are hot, in the B2B space, there's always been what O'Neill calls "Dark APIs", like dark matter because it's happening all around us, but we don't see it. "Perhaps B2B APIs they don't have the hype of fully public APIs, but they are generating very significant revenue."

Of course, another strike against Public APIs is that it can be a public relations gamble. "I believe the more security conscious these organizations are, the less likely they are to start with Public APIs," Higginbotham warned.

Lane is inclined to disagree with his API colleagues. He doesn't believe that having a Public API puts you at a greater risk—because nothing is hacker proof and somebody will eventually find a way to reverse engineer and build on top of it.

“Hiding something is no security strategy. Security through obscurity is no security strategy.”

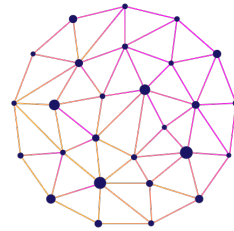
Lane believes that you must have a well-defined [API strategy](#), offering the example of Twitter, which often frustrates developers with security constraints, but nonetheless is a tool that prioritizes its users' security.

Plus, he argues that another positive of publishing a Public API is that you're automatically thinking more about [OAuth and user credential security](#), which will make future growth more easy.

While there is a growing trend of companies pulling back from huge Public API campaigns, the jury is still out if Public or Private APIs will win the future.

Analysis: Are You Getting All You Can out of the API Economy?

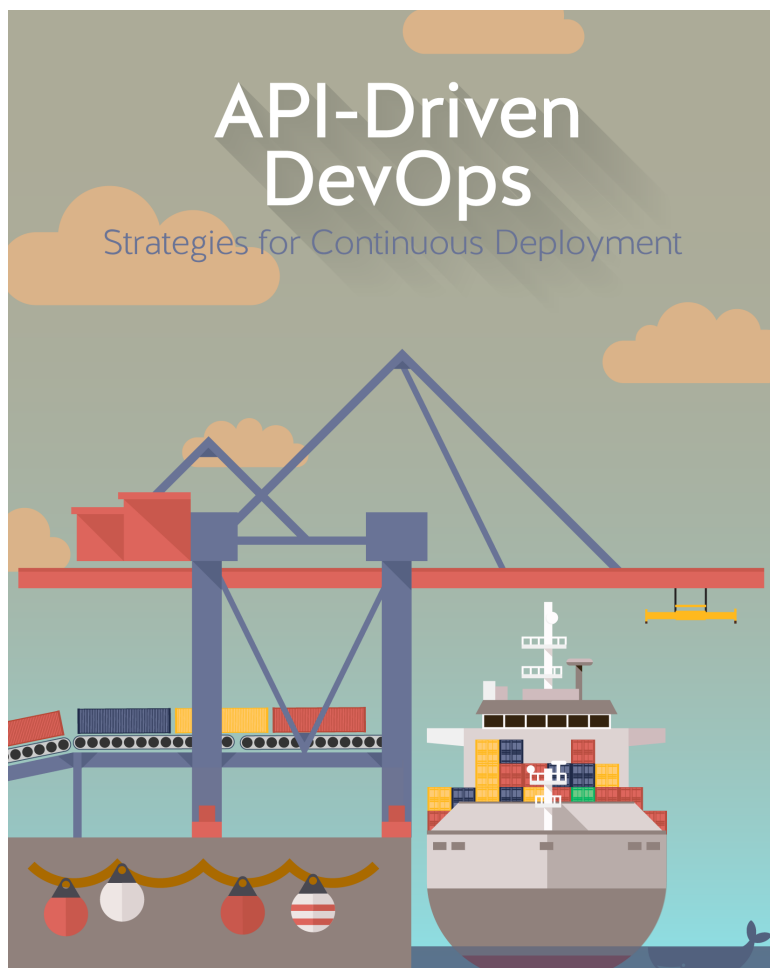
If there's a theme to draw from the insights our API fortune tellers bring, it's that the API isn't for everyone and, like with all things, you need to examine your needs and see what works best for a profitable and productive solution. However, as a developer, we hope we've empowered you with a glimpse into the future of APIs.



Let's review just what that future looks like:

1. APIs allow businesses to become more agile.
2. Microservices architecture helps meet users' demands at a quicker pace.
3. A Public API is not your first move. Private APIs and Partner APIs are more logical to consider before opening yourself up to the world.

Are you ready to be a part of the future of the API space?



AUTHORS

Bill Doerrfeld
Vassili van der Mersch

Kristopher Sandoval
Chris Wood



Follow us to be the first to read next release: [API-Driven DevOps](#).

We hope you enjoyed this volume. We are publishing new content on our [blog](#) every week, as well as hosting [events](#) that bring together API practitioners. Here are additional Nordic APIs resources:



More eBooks by Nordic APIs:

[API-Driven DevOps](#): Grab this volume to learn about the API-driven approach to uniting development and operations.

[The API Lifecycle](#): An agile process for managing the life of an API - the secret sauce to help establish quality standards for all API and microservice providers.

[Programming APIs with the Spark Web Framework](#): Learn how to master Spark Java, a free open source micro framework that can be used to develop powerful APIs alongside JVM-based programming languages.

[Securing the API Stronghold](#): The most comprehensive freely available deep dive into the core tenants of modern web API security, identity control, and access management.

[Developing The API Mindset](#): Distinguishes Public, Private, and Partner API business strategies with use cases from Nordic APIs events.



Nordic APIs Conference Talks

We travel throughout Scandinavia and beyond to host talks to help businesses become more programmable. Be sure to track our [upcoming events](#) if you are ever interested in attending or speaking. Here are some examples of sessions from previous events:

- [Introducing The API Lifecycle, Andreas Krohn](#)
- [You Need An API For That Gadget, Brian Mulloy](#)
- [Pass On Access: User to User Data Sharing With OAuth, Jacob Ideskog](#)
- [APIfying an ERP, Marjukka Niinioja](#)
- [Integrating API Security Into A Comprehensive Identity Platform](#)
- [A Simpler Time: Balancing Simplicity and Complexity, Ronnie Mitra](#)
- [The Nuts and Bolts of API Security: Protecting Your Data at All Times](#)

Endnotes

Nordic APIs is an independent blog and this publication has not been authorized, sponsored, or otherwise approved by any company mentioned in it. All trademarks, servicemarks, registered trademarks, and registered servicemarks are the property of their respective owners.

Nordic APIs AB Box 133 447 24 Vargarda, Sweden

[Facebook](#) | [Twitter](#) | [Linkedin](#) | [Google+](#) | [YouTube](#)

[Blog](#) | [Home](#) | [Newsletter](#) | [Contact](#)