

ProgrammableWeb's 2020 Guide to API Business Models

Deriving ROI from an API Program
on the basis of new revenues and/or
cost reductions



ProgrammableWeb

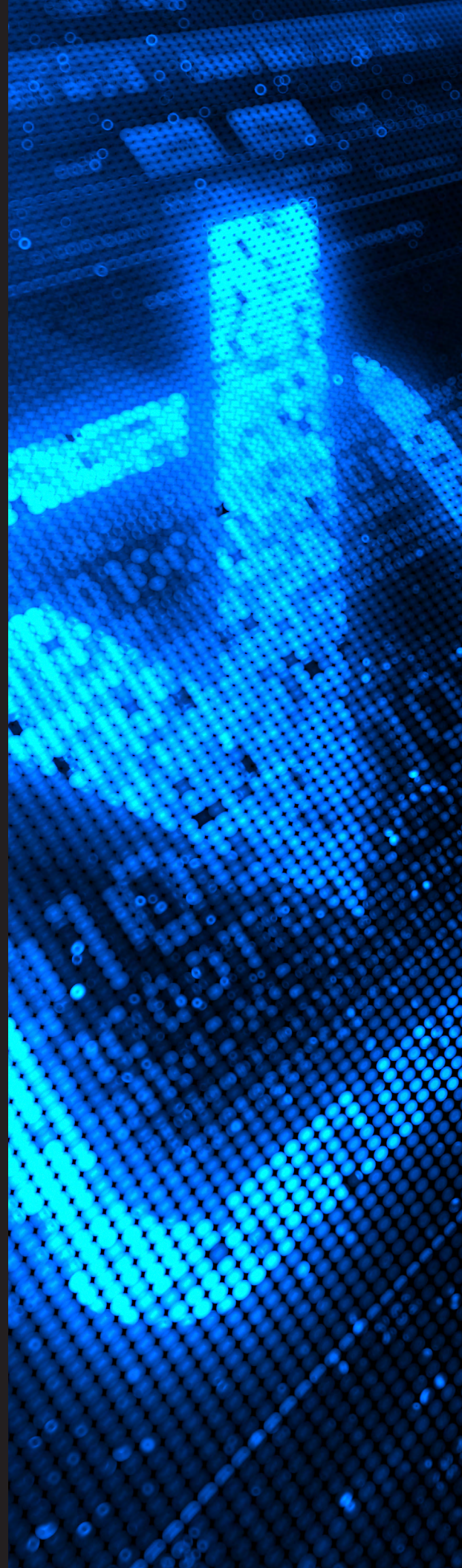


Table of Contents

Preface	4
Introduction	4
Capitalizing on a Digital Strategy	4
State of the API Economy	5
Types of API Business Models	5
Data: The Quid Pro Quo That Cross-Cuts All Business Models	7
Issue of Taxonomy	7
How We've Structured This Guide	8
Terminology	9
Internal Business Models	10
Legacy Modernization	10
Digital Transformation	13
<i>Case Study: Wells Fargo</i>	15
Digital Native	15
<i>Case Study: Airbnb</i>	17
Chargeback	18
External Business Models	20
Partner	21
Revenue Share	21
Affiliate	22
<i>Case Study: Air Malta</i>	23
Resell	24
Ecosystem	25
<i>Case Study: Shutterstock</i>	26
Alliance	27
Consulting	29
<i>Case Study: Acronis</i>	30
Standardization	31
Government-Led	31
Regulatory	32
<i>Case Study: Nordea</i>	33
Influence	34
Industry-Led	36

Productized	37
Upsell	38
<i>Case Study: IQVIA</i>	39
Coin-Operated	40
Key Monetization Attributes	42
<i>Case Study: TomTom</i>	44
Marketing	45
Packaged	45
Audience	47
Brand Loyalty	49
Open API	50
Transparency	51
Community	52
Lead Generation	54
Public Good	56
Outliers	57
Recruiting	57
Alternative Affiliate	57
Directory	58
Mashups	58
 The Other Side of the API Economy	 58
 Methodology For API Business Model Creation	 59
1. Find Potential Value	59
2. Perform Market Research	59
3. Define Unique Value Proposition	60
4. Define The Customer Experience Outcomes	60
5. Establish Partners to Co-Create The Experience	60
6. Discover Expenses Involved	61
7. Build The Developer Experience	61
8. Set Monetization Parameters	61
 Takeaways For API Business Modelling	 62

Preface

By David Berlind

As organizations continue to flesh out their API strategies, one of the biggest questions they look to answer has to do with what API business models they should apply to those strategies. In 2013, *ProgrammableWeb* founder John Musser [published and presented 20 API Business Models in 20 Minutes](#); a body of work that many API economy practitioners still view as the current gold standard guide to API business models.

Over half a decade has passed (an eternity in Internet time) since Musser's initial comprehensive review. With so much having changed across the API economy during that period, *ProgrammableWeb* felt it was time to revisit the question with a brand new comprehensive reference to the many different API business models. In this paper, we've not only endeavored to revisit the old models to see what has changed, but we've also identified all of the newcomers based on real-world observations across the API economy over the last five years.

If there's one thing that definitely has not changed, it's the question that Musser originally said to answer first before asking "What is my business model?" The first question to ask is "What is my business?"

Introduction

Capitalizing on a Digital Strategy

The role of integration is gaining a tremendous foothold. Within today's digital economy, the ability to secure and share data with greater alacrity is modernizing countless enterprises. At the same time, cloud-based microservices are catering to niche developer needs to meet real-time expectations. As the integration value increases, so does accountability. Companies have to defend such connections with a keen digital strategy.

All companies must consider reusability as a means to positively affect profitability (either through improved revenues or reduced costs). The adoption of web APIs is one way to ensure an effective delivery mechanism that also brings future-proofing, platform agnosticism, operational ease and efficiency, and profitable partner connections.

State of the API Economy

In short, the API economy can no longer be shrugged off as a trend. At the time this paper was published, *ProgrammableWeb*, the journal of the API economy and the leading API directory, had chronicled nearly 23,000 public web APIs. As Deloitte says, “APIs are becoming a strategic mandate.” Gartner very simply states: “APIs make digital business work.” APIs are truly the glue that now holds the internet together.

It’s hard to find a business untouched by APIs. If you look to healthcare, eCommerce, AI, travel, IoT, automotive, retail, banking or elsewhere, we find institutions of all types utilizing APIs in some form. And, where organizations are experiencing success with APIs, there’s almost always a well-conceived business model -- some more direct than others -- in place to ensure measurable value to the business. Though APIs are ubiquitous, constructing a working API business model is not as easy as “build it and they will come.” API strategies are full of highly nuanced approaches.

Types of API Business Models

Not all API business models are transparently and **directly** monetized: many benefit the business **indirectly**. For example, many **Internal** APIs are at the forefront of large legacy modernization and digital transformation initiatives, accruing value to the bottom line in the form of saved time and money due in large part to API re-usability.

APIs are also critical to **Partner** connections. APIs are often involved in revenue share agreements, and B2B middleware relies heavily on APIs; each integration can be uniquely priced or valued. Some API standards are designed for entire sectors, with government-led intervention or industry consortiums leading the charge.

Coin-operated APIs — APIs that are billed to the API consumer based on the level of consumption — are frequently productized to serve a business’s operational needs. Example services include payment integration, geolocation, weather, SMS, email, AI, and other programmatic services.

API product owners at the forefront of coin-operated programs utilize various techniques to optimize pricing for unique scenarios and developer consumers. Some, however, do not use direct monetization. Many **Indirect** APIs drive ancillary benefits to the business, such as community building, audience monetization, or brand awareness.

Web APIs have become a major tool for spurring growth. Whether you are constructing an API-first architecture, monetizing an API-as-a-Product, or seeking profitable partner schemes, API business models can be deployed to stay relevant and capitalize on new income opportunities.

In this volume, we’ll attempt to cover all the major types, from private to public-facing strategies:

TYPES OF API BUSINESS MODELS

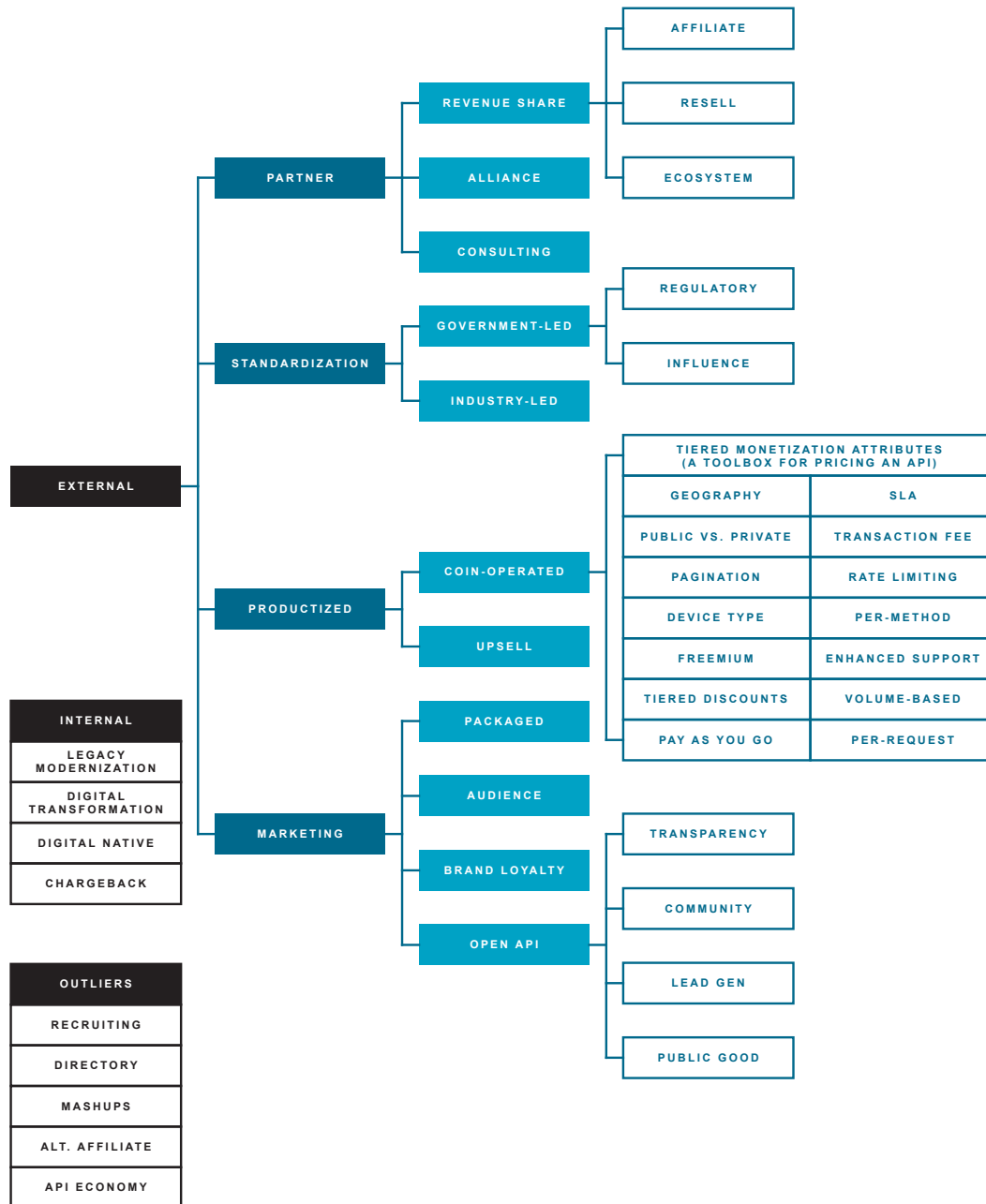


Figure 1: API Business Model diagram — internal, external, and outlier models

We've attempted to design a rubric that is inclusive of current models, and helpful for those entering the API economy. That rubric is represented by the tree hierarchy shown in Figure 1 above. The tree hierarchy attempts to logically organize API business models according to several groupings and sub-groupings. For your monetization conversations, it facilitates important questions like "Should this API be externally exposed outside of our firewall and, if so, should it be productized for consumption by the general public, reserved for partner use, or both?" Using these guidelines, we'll analyze the market to find true business worth in the secure provisioning of data and functionality. Naturally, companies may adopt attributes from various models to compose a holistic platform strategy. Whereas some of an organization's APIs might follow one model, other APIs can follow another. There's no one size fits all and if we've learned anything from the ever-evolving API economy, it's that new models will keep coming.

Data: The Quid Pro Quo That Cross-Cuts All Business Models

There's an old saying dating back to 1970's broadcast television that said "if you're not the customer, you're the product." Over 40 years later, the saying couldn't be more true when it comes to the Web and services like Facebook for which we pay nothing.

But when it comes to the API economy, this old axiom could be expanded to say "Even when you're the customer, you're the product too." In other words, regardless of whether money exchanges hands as the result of an API interaction, the API provider is always getting new data as a result of that API interaction. In some cases, it may be nothing more than a glimpse of how developers are using your API. What are their favorite resources to call? What are the common syntactical mistakes. In other cases, it might be like what happens every time a API call is made to the Google Maps API; Google learns a little bit more about who is looking for what destination.

What you, as an API provider, do with that data once you've recorded it in your databases and how you derive additional value from it is up to you. So, as you sift through the many business models that we've enumerated in this paper, be sure to keep in mind that this is one quid pro quo between developers and API providers that cuts across all of them.

Issue of Taxonomy

When forming our classification system, we tried to balance granularity with magnification. We have attempted to only create boundaries around API models we've observed across the API economy. However, due to the nature of private company information, some taxonomies are educated guesses.

It should also be noted that an API provider may utilize many different API models within its digital strategy. Some models naturally overlap with others. For example, an Audience API defined in the Marketing category may monetize some

operations through Coin-Operated Productization. The same platform may utilize Crowdsourcing within its Open API projects, and so on. Though not all taxonomies are as clear cut as the business reality, we feel this text will supply your business with a common language to talk and compare various strategies.

Lastly, the API Economy is still new. It is ever-evolving. Throughout our research, we found outlying models that were too nascent or unproven to warrant inclusion within this taxonomy. We have identified these models within the **Outlier** section.

How We've Structured This Guide

Before diving in, here's how this report is organized to make it easy for you to read. The different sections of this report follow the branches of our business model hierarchy shown in figure 1 above. For example, the first main section below covers the first of the hierarchy's two main branches -- the Internal Business Models branch -- and each of its four leaves; Legacy Modernization, Digital Transformation, Digital Native and Chargeback. But in the second branch of the hierarchy's two main branches -- the External Business Models branch -- there is additional branching before we get down to the leaves (the actual business models themselves).

At the branch level, our paper includes some narrative about what distinguishes the branch you're reading about from its sibling branches. But, as you will see, at the leaf-level, we not only offer some narrative about the business model itself, we also use a standard, easy to scan table structure to characterize each leaf, where it fits in to the overall hierarchy, some of its main distinguishing factors, the type of developing consumer it targets, the industries it is well suited to, and examples of organizations putting it to use. For some business models, we even provide a short case study of the business model in practice. As we navigate the branches and leaves of the hierarchy, we rely on some standard terminology that you should be familiar with.

Terminology

<i>API</i>	Stands for Application Programming Interface. When discussing API business models we typically mean web APIs.
<i>API consumer</i>	The developers and clients that call the API and integrate it into their applications.
<i>API provider</i>	The company that creates and exposes an API for developers to use.
<i>API business model</i>	A method to make a Return On Investment (ROI) from providing an API. There are many direct and indirect business benefits, including external and internal use cases for providing APIs.
<i>Developer Experience (DX)</i>	The experience a developer has when consuming an API. It is a good practice to decrease friction when consuming APIs, have support guides, and offer ongoing maintenance. Reliable and usable APIs have great developer experiences. For your reference, <i>ProgrammableWeb</i> has authored a comprehensive guide to building world-class developer experiences should you want to know more about this topic.
<i>API as a product</i>	Once, APIs were just seen as technical tools. Now, they are big business. A recent IT business trend is to treat the API as a product in its own right. This implies added emphasis on business development and marketing.
<i>Direct</i>	Models that are clear revenue-generating activities. There is a direct correlation between API consumption and revenue generation.
<i>Indirect</i>	Models that positively affect elements of the business, which in turn may increase net revenue. Indirect models include expense saving initiatives.

Internal Business Models

As can be seen from Figure 1, API business models are divided into two major branches: Internal and External. With the rise of digitalization, mobile, and cloud technologies, nearly all companies are undergoing an internal transformation process. And if your company isn't, it probably should be. This may take the form of **Legacy Modernization**, in which monolithic architectures are decomposed into smaller, more serviceable components that lead to other significant efficiencies. Large brands are also undergoing **Digital Transformation**, reinventing their traditionally analog customer experiences and processes into ones that are much more digital in nature (or at least digitally driven). In both cases, some enterprises adopt a **Chargeback Model** to monitor tech usage. For some organizations, this could be key to avoiding bloat when introducing any new technologies.

Within internal digitization initiatives, APIs are a fundamental component. They help streamline internal operations, enabling microservice fluidity and act as a single source of truth for a holistic platform mindset (so long as there are no workarounds or circumventions of the APIs). When such APIs drive internal development, they are typically private to the organization's applications and processes (often behind a corporate firewall), and not publicly documented. Nonetheless, API-driven internal initiatives may also act as an **R&D Investment**; a test-bed for innovative technology and a foundation for future externalization.

According to Gartner, organizations that operate in this way are said to be operating "bimodally." APIs that are developed to support internal legacy modernization or digital transformation initiatives involving greater predictability and existing systems of record are said to be operating in Mode 1 of Gartner's bimodal model. But when that same work serves as an R&D testbed for areas requiring exploration — for example, exposing those same APIs to developers outside of the organization with something more speculative in mind — Gartner defines this as Mode 2 activity of its bimodal model. Are you primarily attracted to Mode 1? Mode 2? Or both?

Legacy Modernization

The world is changing, becoming more modern and digitized by the millisecond. But for large IT systems, the digital backbone isn't always pretty. Layers upon layers of legacy infrastructure can lead to massive technical debt and scalability issues while impeding the aggregation of disparate data into various 360-degree views (ie: of the customer). Adaptation for such enterprise software infrastructure is difficult. Misaligned data structures, lack of accessibility, opposing standards, shadow IT and other issues can become roadblocks to corporate innovation and agility.

Legacy Modernization thus seeks to refactor traditional software infrastructure by taking a pre-existing monolithic architecture and making it more serviceable. It is the goal of such efforts to reimagine Gartner's aforementioned Mode 1 infrastructure in a more modular way. Such modernization efforts continually design reusable

systems with API-led componentry. Unilever, Airbus, Wells Fargo, and many other institutions utilize APIs within its legacy modernization efforts to great effect.

Legacy Modernization is an indirect API business model that can benefit an organization in three key ways: by reducing costs, creating reusable components, and improving efficiency.

- **Reducing Costs:** Traditional monolithic hardwired processes may involve multiple point-to-point connected systems. Modernizing the relationships between such systems with an API contract can have a dramatic effect. When architectures are viewed independently, more cost-effective substitutions can be made. For example, if a system mainframe is wrapped in a standard API contract, another system can be substituted for that mainframe that conforms to the same API contract. Reconstituting existing expensive systems with Amazon cloud technology is an example of this interoperability and substitution in practice.
- **Reusable Components:** Once elements of your IT infrastructure are exposed through APIs, it becomes easier to compose new applications from those APIs as though they are reusable building blocks, yielding what is known as a highly composable enterprise. Removing the need for redundant code decreases production efforts and in net, decreases costs.
- **Improved Efficiency:** API-first modernization improves the business's ability to respond to new pressures and innovate more efficiently. It enables business leaders to more easily make tech substitutions for other reasons not directly associated with cost (eg: performance, security, reliability, etc.). With API-driven microservices, business is able to change and iterate, thus increasing agility.

<i>Taxonomy</i>	Internal > Legacy Modernization
<i>Type</i>	Internal
<i>Definition</i>	Legacy Modernization seeks to revolutionize pre-existing monolithic software by breaking it into more serviceable API-driven components.
<i>Business Model</i>	Reduce costs by creating reusable components, improving efficiency and enabling cost-effective substitutions.

<i>Developer Consumer</i>	Internal corporate developer
<i>Key Industries</i>	Internal systems, enterprise architecture, large corporate environments.
<i>Examples</i>	<ul style="list-style-type: none"> • Siemens: Siemens, the large manufacturing and electronics company, has modernized using APIs to share data between previously siloed mainframes. With this new strategy, Siemens has increased project delivery speed by 50% and has enabled external exposure to partners, regulators, and new user-facing experiences. • Airbus: To overcome development hurdles, Airbus has sought to become a composable enterprise. This meant modernizing their backend with reusable components to unlock ERP data and self-service options to accelerate the manufacturing and supply chain. “We are moving to composable building blocks, microservices, and APIs. It’s about developing faster and cheaper, and at scale,” says Chris Taylor, VP, Digital Accelerator, Airbus. • British Petroleum (BP): Through the help of a modernized backend, BP is finding leaner ways to produce oil while reducing emissions. “Our API strategy is right at the heart of our application ambition and technology transformation,” says Diana Kennedy, VP, IT&S Strategy, Architecture and Planning, BP. <p>Others organizations that have undertaken major legacy modernization initiatives include:</p> <ul style="list-style-type: none"> HSBC Pilot Fyling J Splunk State of Colorado Addison Lee New Relic

Digital Transformation

Many sectors are reinventing themselves in the wake of new digital experiences and expectations for user-facing applications. By empowering brands with digital experiences, historically analog services are finding new markets and novel ways to engage with customers.

Digital Transformation transforms traditional analog processes and customer-facing experiences by introducing new digital experiences. Within many digital transformation initiatives, APIs are utilized as a means to facilitate better communication between new modular components, transforming business functions to be more digitally-driven. Not only is digital transformation helpful, but necessary for revenue retention. According to Mulesoft's 2018 Connectivity Benchmark Report, "four out of five IT leaders (81%) say that they will lose revenue in as little as 12 months if digital transformation isn't completed."

The business model for Digital Transformation is indirect. API-based digital transformation brings cost savings, improved agility, and sets the foundation for newfound customer-facing products and services which, in turn, can result in new channels of business or improved revenues. Business benefits from digital transformation include:

- **Modernization Benefits:** Digital transformation shares many of the same benefits with legacy modernization in terms of cost-savings, reusability, and efficiency. It helps fill IT delivery gaps, reducing efforts involved with pushing new services to market.
- **Quick User Experiences:** Once digital transformation is API-driven, enterprises can build out and iterate user experiences more quickly. Marketing campaigns can test and fail fast. APIs may be dogfooded to distribute data and functionality across devices over the web, agnostic of the platform.
- **New Customer Experiences:** API integrations enable innovative customer experiences. Companies can mashup new internal APIs (even including APIs from other organizations), and imagine new lines of revenue or loyalty programs.
- **New Products:** With API-based digital transformation, a CIO is truly a Chief Imagination Officer. Not only are new experiences around existing products possible, but by thinking digitally, CIOs can piece together entirely new services and products that enhance customer interactions with the brand.

<i>Taxonomy</i>	Internal > Digital Transformation
<i>Type</i>	Indirect

<i>Definition</i>	Digital Transformation transforms traditional analog customer-facing experiences by introducing new digital experiences. APIs are utilized as a means to facilitate better communication between new modular components and transform business functions to be digitally driven.
<i>Business Model</i>	API-driven digital transformation brings cost savings, improved agility, and sets the foundation for newfound customer-facing products and services which in turn can result in new channels of business or improved revenues.
<i>Developer Consumer</i>	Internal developer
<i>Key Industries</i>	Consumer brands, eCommerce, Retail.
<i>Examples</i>	<ul style="list-style-type: none"> • Unilever: Using APIs, Unilever modernized its eCommerce portfolio with common operating procedures across its many user-facing brands. Uniting these global identities with a common framework meant a unified experience across Unilever's products. • McDonald's: With mobile ordering, digital kiosks, and streamlined table service, McDonald's is rethinking the way the brand transforms by utilizing digital services. APIs are part and parcel of McDonald's goal to become a "modern, progressive burger company." Such API services are driving customer-facing applications for mobile, social, web, and restaurant-based kiosks. • New York & Company: APIs are helping the large retailer New York & Company unify its customer experience across many digital touchpoints. Its API-led approach enables real-time inventory queries, retail kiosks with end-to-end transactions, and additional omnichannel customer relationships. <p>Others organizations that have gone through significant digital transformations include:</p> <p>Wells Fargo</p> <p>Levis</p> <p>Salesforce</p>

Case Study: Wells Fargo

Wells Fargo has roughly 273,000 employees and serves 70 million customers across 8,500 locations. As part of its digital transformation initiative, the bank built the Wells Fargo Gateway platform to deliver a unified digital experience, whether over the phone, online, or through mobile devices. Through these various channels of engagement, an API-led approach to connectivity allows the bank to integrate its disparate applications to deliver a more responsive and robust experience for both customers and partners. The Wells Fargo API conduit supports a large digital platform, increasing the speed of IT delivery and helping Wells Fargo respond quicker to new user demands.

Digital Native

From startups to unicorns, newer companies have bet on digital strategies since their origin. These companies with digital native roots, bred in the mid-2000s through 2010s, innately understand the pressures of mobile, big data, and web integrations. Many platform models we utilize today such as Uber, Spotify, and Amazon are digital native, with a storied record of leveraging internal APIs to accelerate disruption within their sectors.

Digital Native companies use APIs to power their apps and create consistent experiences across all devices. These digitally born, API-first organizations have benefited from a platform-agnostic web service since their early beginnings. Such APIs act as an internal single source of truth. Having utilized digital strata from the onset, Digital Native businesses don't explicitly fit into the Legacy Modernization or Digital Transformation categories, which both seek to revive existing components.

Digital Native is an indirect API business model. Digital Native companies heed the lessons learned from previous generations of monolithic IT and, in an effort not to repeat history, utilize APIs to avoid costs otherwise incurred in non-API led approaches. This can also improve decomposability; the act of separating functionalities into microservices continues to be paired with the act of preserving platform longevity without incurring debilitating long term technical debt. With this optimization in place, API-first digital native businesses retain a strong competitive edge over businesses that are weighed down by legacy technical approaches. In many ways, organizations that undertake major legacy modernizations and digital transformations are looking to achieve a digitally native state.

- **Avoid Costs:** Digital native companies utilize APIs to avoid unnecessary costs. With APIs, data distribution is streamlined to deliver a consistent experience across all devices.
- **Optimize:** Such an API-based stack has a high level of reusability. Digital native groups utilize APIs to continuously improve organizational and systems efficiency. With plug and play capability, microservices architectures are repurposed within many environments, and usage is optimized for specific scenarios.

- **Competitive Edge:** The agility gleaned from API-first integrations enable digital native companies to disrupt entrenched competitors that are tied to traditional technological stacks.

Many digital native API-first platforms have flourished into large enterprises. For example, **Spotify** utilizes an internal API to distribute content to all end-user applications. It also utilizes internal Payment APIs to handle payment processing for a variety of transaction types. At Uber, APIs provide high functionality with lightweight engineering. Uber's smart integrations enable navigation, sending messages, and payments — all without forcing Uber to create those services on its own.

For Amazon, being digitally native has meant being API driven across its entire stack; covering system APIs (APIs on systems of record), process APIs (APIs fueled through aggregation of systems APIs and any subsequent transformations), and experience APIs that drive the experiences seen (or heard!) by the ultimate consumers (customers, employees, etc). This digitally native approach allows Amazon to innovate new customer experiences at lightning speed. For example, a microwave oven with a button for replenishing a customer's supply of microwavable popcorn could be invented in days as opposed to months or years. Such functionality is driven by Amazon's existing APIs for product replenishment, Alexa, and shopping.

<i>Taxonomy</i>	Internal > Digital Transformation
<i>Type</i>	Indirect
<i>Definition</i>	Digital Native businesses utilize APIs at their core. Such API-first platforms epitomize reusability, streamlined data-sharing approaches, and microservice-driven architectures and organizations.
<i>Business Model</i>	Digital Native companies utilize APIs to avoid unnecessary costs. In comparison to competitors, they utilize APIs to optimize for efficient system utilization and minimal waste. Digital native companies may disrupt competitors who are weighed down by legacy technology, simultaneously delivering better and consistent experiences across all devices.
<i>Developer Consumer</i>	Internal developer

<i>Key Industries</i>	Media, Transportation, Entertainment, Cloud, Infrastructure, SaaS
<i>Examples</i>	<ul style="list-style-type: none"> • Spotify: Spotify, the world's largest music streaming platform, has adopted APIs as part of their core strategy since day one. Their internal APIs can distribute content to any end-user device. They also built internal Payment APIs to help accept hundreds of currency and digital payment options. Both initiatives have a direct positive net effect on consumer satisfaction which in turn influences new and renewed subscriptions. • Netflix: The Netflix internal API is a model private API. This internal API acts as the backbone for their quick video services, providing a data transfer layer agnostic of the client device. It also enabled easier partner integrations. Netflix's model closely resembles the layered system, process, and experience approach to API architecture. This technology was critical for creating easy consumer usability that led to Netflix's disruption of the media industry. • Amazon: Amazon's corporate motto is to develop all digital services with externalization in mind from the start. Thanks to this approach, Amazon develops and reuses hundreds of components, helping streamline internal development efforts to create robust, smart user experiences. Many internal services, such as AWS, have inspired externalized products. <p>Other organizations that are digital natives include:</p> <p>Airbnb</p> <p>Uber</p> <p>Roku</p>

Case Study: Airbnb

Airbnb has been a mobile-first digital native from day one. For a company experiencing such tremendous growth, the technological demands have naturally changed as usage demands have skyrocketed and its mobile facades improve. By adopting an API-first mentality, Airbnb has decreased platform-wide inconsistencies with development consolidation, benefiting overall internal operational efficiency.

Airbnb's original API v1 was constructed as a huge monolithic Ruby on Rails application, aptly named Monorail. Originally, an internal API powered its mobile apps, and a web controller powered the website. Now, the company is pivoting toward adopting a single source of truth; an internal Graphist API that distributes Airbnb content to all platforms.

For Airbnb, using a standard company-wide API framework ensures various teams construct APIs in the same way; like a style guide for the enterprise. Its approach also involves explicit whitelist and blacklist checks to ensure no malformed requests are made. It also enables performance-boosting features like deferred loads and batch aggregation. With thousands of endpoints and dozens of developers building on it, Graphist is steadily growing in favor.

Embed: APIs as The Source of Truth, Zane Claes, Nordic APIs.

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/qqv3s_c_5Jc" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
```

Chargeback

IT is traditionally viewed as a cost center, often treated as overhead from an accounting point of view. Thus, departments often get charged-back for their consumption of IT resources through some sort of allocation model (ie. overhead percentage, per-use). Les Johnson of Cella defines chargeback as "running your department in the most efficient manner to support maximizing your company's profits."

Logistically, **Chargeback Models** come in many forms. IT chargebacks may be viewed as overhead, in which the provided IT systems cut across an entire company. Or, chargebacks may be based on the utilization of resources belonging to an internal IT department. When IT infrastructure is delivered through API connections, it introduces a convenient business opportunity to monitor tech usage based on API consumption metrics (which is arguably more accurate than most overhead models). Such a practice requires usage reporting and an automated platform for self-service. Interestingly, the unit-based pricing models of chargeback programs are strikingly similar to coin-operated API monetization models. This is especially true in some cases where a specific IT resource (people, machines, software, etc) is tied directly to an internal consumer.

Chargeback Models benefit the business indirectly by improving accounting

procedures, improving visibility into technical debt, and demonstrating the cost-benefit of APIs. Utilizing APIs for initiating a chargeback model can be extremely helpful, unlocking benefits such as:

- **Visibility:** Provides useful metrics about business operations. Increased awareness of how much IT bandwidth application development requires.
- **Demonstrate Value:** Utilizing chargeback models, organizations can demonstrate the value and cost-benefit of their APIs. Such testing grounds can prove the financial efficacy of having APIs in the first place.
- **Accounting:** API-driven chargeback systems can reduce overall accounting department costs. Streamlining with APIs may alleviate the need for bulky systems, reducing technical debt.
- **Communication:** Transparency helps interdepartmental relationships. Monitoring how groups utilize services is integral for communication, as well as ongoing support and maintenance.
- **Prioritization:** When organizations move to a consumption-based chargeback model, they should be able to more easily spot areas of growth and prioritize resources accordingly.

<i>Taxonomy</i>	Internal > Chargeback Model
<i>Type</i>	Semi-Direct
<i>Definition</i>	A Chargeback Model is when the IT department charges internal consumers (developers in the case where chargeback is for API consumption) for the use of IT architecture. In doing so, you are able to more accurately quantify departmental consumption of IT resources.
<i>Business Model</i>	An internal version of productization that can help streamline accounting across an organization. Departments may be charged on a per-use basis. Grants better visibility into the value of services, tech usage.
<i>Developer Consumer</i>	Accountants, Finance Engineers, Internal Developers

<i>Key Areas</i>	Enterprise Accounting, Large Organizations Accounting, large corporate environments, inter-departmental communication.
<i>Examples</i>	<ul style="list-style-type: none"> • Cloudability: Cloudability provides SaaS chargeback services. • Cube Billing: Cube Billing provides a Software-as-a-Service IT Cost Allocation and Chargeback/Showback Software. • IBM: IBM's SmartCloud Cost Management is an IT system for chargeback models. <p>Other providers that offer chargeback solutions:</p> <ul style="list-style-type: none"> Chargeback API DellEMC, Storage Resource Manager (SRM) VMware, vRealize Business Red Hat, Red Hat CloudForms Talligent, Openbook Platform Embotics, vCommander Cloud Management Platform ClearCost Elements CloudBolt, Hybrid cloud management platform developed by CloudBolt Software Azure Cost Management + Billing Open iT, Open iT for Software Asset Management and Chargeback

External Business Models

More often than not, when organizations consider the idea of joining the so-called API Economy, they are talking about joining the growing list of organizations that have externalized their APIs beyond their corporate firewalls for third-party consumption. Collectively, these external APIs from all these different organizations (a.k.a. "API Providers") from all around the world form the basis of a global API Economy. However, it should be noted that there are several types of third parties and it's not as simple as saying that an external API is a public API as many pundits are wont to do. This is why, at the top level of our hierarchy, we've divided the world

between internal and external APIs rather than private and public APIs. Put another way, all public APIs are external. But not all external APIs are public.

At the very moment an organization starts to consider externalizing an API, it is usually doing so with the idea of opening up new, digitally-driven, business or revenue opportunities which in turn is often viewed under the larger rubric of monetizing APIs. However, the idea of monetization is often misconstrued to mean one thing; ringing the cash register according to level of API consumption. Indeed, such “coin-operated” APIs (discussed later in this paper) are a form of API monetization.

As implied in our coverage of Internal Business Models, ProgrammableWeb holds the view that any time an organization is improving its financial bottom line due to the implementation of APIs -- be they internal or external -- that organization is essentially monetizing its APIs. But for organizations that really want to understand their options when it comes to monetizing APIs through externalization, the rest of this paper breaks it down for you.

Partner

Partner programs involve API integrations with select partners. They are more open than strictly internal APIs, however, they are not so open that they become self-service public APIs. Partner agreements involve carefully vetted partners and tend to be less publicly transparent. For organizations contemplating the idea of API monetization, partner programs may be the first logical step after their Legacy Modernization or Digital Transformation is well underway, opening up a new economy of potential user experiences and revenue-generating tactics.

Partner programs involve three sub-groups of business models; revenue share, consulting, and custom platform integrations. **Revenue Share** business models can differ in style and purpose. API providers may also seek to offer fee-based **Consulting** to help consumers use their APIs. Lastly, **Platform** APIs are true partner integrations. Emerging out of necessity or tech alignment, Platform partner APIs often advance the core competencies of both provider and consumer.

Revenue Share

In Revenue Share agreements, an API provider opens up API access to select partners who incorporate the API(s) into their applications. When end-customers utilize the partner API connection to initiate a purchase (or as part of an ongoing subscription), the API provider grants the partner a certain amount of the transaction per sale in the form of a commission or percentage. We've identified three nuanced styles of Revenue Share agreements: **Affiliate**, **Resell**, and **Ecosystem**.

Affiliate

Affiliate is a type of Revenue Share partnership in which the API provider allows partners to resell through the API as part of an affiliate network. Such affiliate campaigns give rewards to partners who utilize the API to showcase the API provider's products or services. The partner developer receives a small share of any revenue. Compared to other revenue share agreements, affiliate programs are more standardized and self-service, typically not processed on a negotiated or customized basis. Affiliate programs may also pay developers based on clicks and conversion rates, as opposed to strictly sales.

<i>Taxonomy</i>	External > Partner > Revenue Share > Affiliate
<i>Type</i>	Direct
<i>Definition</i>	API provider enables third-party developers to utilize API connections to increase marketing and sales for its products or services.
<i>Business Model</i>	The API provider opens up new avenues of revenue, splitting proceeds with partners through percentage conversion rates, click-throughs, or other affiliate network criteria.
<i>Developer Consumer</i>	Independent Software Vendors (ISV), mobile apps, online shops, affiliate marketing developers.
<i>Key Areas</i>	eCommerce, Transportation, Air Travel, Hotels, Online bookings, Retail
<i>Examples</i>	<ul style="list-style-type: none">• eBay Partner Network (EPN): As part of eBay's API Developer Program, eBay's partner network offers revenue whenever you send someone to eBay and they make a purchase within 24 hours. Depending on the category of item, partners earn between 50% and 70% of eBay revenue. EPN provides many affiliate resources, including link builders, smart placements, creative generators, and more. The EPN APIs offer the most programmatic customized ability. The APIs available within the eBay Developers Program include Sell APIs, Buy APIs, Search APIs, After Sale APIs, and many more.

Examples (cont'd)

- **Walgreens Photo Print API:** The [Walgreens Photo Print API](#) enables end-users of applications built by third-party developers to order on-demand photo prints from Walgreens locations. Developers, such as Pictarine, take up to 20% in revenue.

- **Addison Lee:** London-based car service provides revenue sharing benefits from [API-enabled purchases](#) that utilize deep links in third-party apps.

Others companies that offer a revenue share to developers on an affiliate bases include:

[Air Malta](#)

[Amazon Product Advertising API](#)

[MakCorps](#)

[Skyscanner Travel APIs](#)

[American Airlines NDC](#)

[Shutterstock Affiliate Program](#)

Case Study: Air Malta

In the travel industry, integration is key. Air Malta, “the airline of the Maltese Islands,” is a 10 fleet airline servicing Europe, the Middle East, and North Africa regions. Though its size is small, its recent Digital Transformation efforts have opened up new profitable opportunities. Air Malta utilizes a hybrid IT model, exposing some of the API-first services they use internally (searching, booking, tickets, facilities) to third parties as well.

By externalizing their flight booking web services, Air Malta has increased sales by distributing data via a partner affiliate model. Not only have partner APIs for automated booking increased overall revenue at Air Malta, it has become a competitive advantage. For example, in allowing partners such as RyanAir to access Air Malta’s interior infrastructure, Air Malta is able to compete at the same level as other larger cloud-based carriers. Furthermore, the organization’s API-driven development process drives down costs through internal reusability and efficiency.

Resell

Some partner programs involve reselling APIs themselves. **Resell** is a type of Partner API business model, a commission-based transaction where an API provider works with partner companies to resell the provider's API(s) to its customers and partners. The partner thus takes a percentage of the sale with each integration on a negotiated basis. This is unique from Affiliate as the API itself is being resold by partners to third parties.

<i>Taxonomy</i>	External > Partner > Revenue Share > Affiliate
<i>Type</i>	Direct
<i>Definition</i>	API provider enables third-party developers to utilize API connections to increase marketing and sales for its products or services.
<i>Business Model</i>	The API provider opens up new avenues of revenue, splitting proceeds with partners through percentage conversion rates, click-throughs, or other affiliate network criteria.
<i>Developer Consumer</i>	Independent Software Vendors (ISV), mobile apps, online shops, affiliate marketing developers.
<i>Key Areas</i>	Platform-as-a-Service (PaaS), Payments, eCommerce
<i>Examples</i>	<ul style="list-style-type: none">• Consulting Firms: Large consulting firms such as Deloitte, Accenture, or Capgemini with giant technology contracts may resell APIs within these dealings. API providers may partner with a consultant group such as Deloitte to expose their API to the consulting group's customers.• Dwolla Reseller Partnership: For their Reseller Partners, Dwolla offers a white-labeled bank transfer API that manages customers, verifies bank accounts and facilitates payments. The Dwolla Integration program enables the reselling of Dwolla infrastructure, which includes an application suite and APIs. Productfy is an example of a reseller of Dwolla services. Productfy utilizes the Dwolla API to offer a toolkit for building financial applications, enabling things like bank verification, ACH payments, and user identification.

Examples (cont'd)

- **RapidAPI:** In a way, the [RapidAPI marketplace](#) resells APIs. By placing a developer experience shim on top of API services, RapidAPI is able to resell and share revenue with the providers of those API services.

Ecosystem

Sometimes, providers expose an entire platform that may hold embedded APIs. Such platforms may also utilize a marketplace model in which revenue share is split between the provider and third parties. This model warrants its own taxonomy as such dealings are larger in scope than Affiliate models or Reseller programs.

Ecosystem is a type of Partner API model in which The API provider exposes its platform so partners may capitalize on an ecosystem of connections and integrations to other solutions and services. The big difference here is that partners resell **entire applications**, which in turn, have embedded APIs. The API provider thus strikes a custom deal with partner developers based on revenue sharing.

<i>Taxonomy</i>	External > Partner > Revenue Share > Ecosystem
<i>Type</i>	Direct
<i>Definition</i>	A provider exposes an entire platform to third parties, selling entire applications that utilize APIs.
<i>Definition</i>	Consulting Firms: Large consulting firms such as Deloitte, Accenture, or Capgemini with giant technology contracts may resell APIs within these dealings. API providers may partner with a consultant group such as Deloitte to expose their API to the consulting group's customers.
<i>Developer Consumer</i>	Systems Integrators, application resellers
<i>Key Areas</i>	Infrastructure-as-a-Service (IaaS), Business-as-a-Platform (BaaS), Systems Integrators

Examples

- **Intuit Quickbook API Ecosystem:** When accountants are involved as resellers in the deployment, the business model shifts to one based on revenue sharing. “Accountants have the ability to resell third-party apps that are integrated into the platform and present a single bill to the customer,” said Alex Barnett, head of Intuit’s Developer Group (see [How Intuit Designed its API Ecosystem For Both Internal and External Success](#)).
- **Odin APS Package:** [The Odin APS Package](#) (formerly Profitbricks) provides a “metered server and networking infrastructure” to its clients. This package allows resellers to provide cloud hosting services to their users. A service provider utilizes their ecosystem to create a package that acts as the middleman between resellers and users. The package utilizes an API within this environment to provision resources.
- **Salesforce Lightning:** The developer ecosystem around the Salesforce platform includes third-party-built turnkey software components (known as [Lightning components](#)) that are sold through a Salesforce operated marketplace called [AppExchange](#). Through Salesforce’s AppExchange Partner Program, Salesforce offers partners and developers various opportunities to market those wares. The Program grants a share of their revenue, and benefits from the reach and awareness the marketplace drives. The Salesforce developer platform offers custom-tailored Salesforce experiences through Lightning components and SDKs purpose-built for working with specific APIs.

Case Study: Shutterstock

For an example of Partner API models in practice, take Shutterstock. Shutterstock is a stock media supplier, offering APIs to enable easy partner integrations. Their API partners include print-on-demand tools, social platforms, and web design tools for users in need of quick content.

API integrations are the means by which Shutterstock embeds native content licensing into the Facebook Ads creation panel. Thanks to this integration, users can search and select free stock images within the Facebook Ad experience. Shutterstock charges Facebook at a subsidized price to license these graphics and video.

Shutterstock also has Revenue Share programs with selected partners. One example is Wix, the online web builder tool. On Wix, Shutterstock content and footage are

natively embedded. The user pays for Shutterstock content within the Wix platform, and Shutterstock grants Wix a certain percentage of the sale.

Shutterstock is also experimenting with direct API monetization, externalizing their internal tools. The algorithms they have developed to maintain their own content databases, such as auto-tagging for nudity or gun detection, are now being resold for external censorship and content moderation needs. Revaz Tsivtsivadze, Head of Product, Shutterstock Developer Platform, estimates that API monetization now accounts for 5%-10% of overall revenue, a figure which is steadily growing.

Shutterstock also supports an Affiliate Program for more public needs; widgets that link to Shutterstock. Shutterstock is a jack of many API business models, utilizing licensing, revenue share, and productization through direct monetization — proof that a holistically conceived API program has the potential to open up new, accretive channels of revenue for an organization, each of which involves different business models.

Alliance

For some Partner API business models, Revenue Sharing is not a driving factor behind integration. Instead, two companies (or more) may seek to partner through API connections to advance shared goals.

An **Alliance** model is thus a partner-to-partner integration that is complementary, advancing the core competencies of both parties involved. Logistically, Platform partner business models may involve customized development. Both parties benefit from a custom API-based integration that drives the success of both organizations. Partners in Platform API programs may serve a mutual customer.

An automotive company, for example, may encourage partners to integrate with their connected dashboard to provide advanced functionality, both improving end-user experiences and doing so in a lean way that outsources software development.

<i>Taxonomy</i>	External > Partner > Alliance
<i>Type</i>	Indirect
<i>Definition</i>	The API provider opens integration capabilities without direct monetary goals. This is different than open APIs as Platform models are typically on a negotiated basis. Alliance models do not involve revenue share. Instead, the collaboration is mutually beneficial to the success of both parties.

<i>Business Model</i>	The business opens integrations with a partner to advance complementary facets to serve mutual customers, driving success for both organizations.
<i>Developer Consumer</i>	Partner developers with close ties to business objectives
<i>Key Areas</i>	IoT Devices, Entertainment, Media, Streaming, Smart TV, Automotive, IIoT, Smart Home technology
<i>Examples</i>	<ul style="list-style-type: none"> • Smart TVs: Smart devices require media, and media symbiotically requires devices. Thus, an API is a logical way for such platforms to establish a base and to offer mutual benefits. For example, Samsung Smart TV offers a Samsung Product API to access features specific to Samsung TVs. Similar cases include Sony's Audio Control API and its Playstation Formats Partner Agreement. • Automotive: With complex digital components like dashboards and onboard sensors, cars produce more data than ever before. The connected nature of cars necessitates APIs for third-party partnerships, enabling fleet management, telematics, and other use cases. This data could also help inform compliance, manufacturing, or power innovative user experiences. Example partner programs in the automotive industry include Daimler's oneAPI{HUB} or BMW Smartcar's API. Ford Developer Program also offers AppLink, "a suite of APIs that provide the ability for mobile developers to extend the command and control of a mobile application to the in-vehicle Human Machine Interface (HMI)." • Supply Chain Logistics: Shipping and logistics companies utilize standard digital protocols for their container tracking systems. EDI was historically the data format of choice within the supply chain industry. APIs are now enabling B2B supply chain partners to transmit data more fluidly and are allowing secondary actions to act upon data. Maersk Digital Solutions, for example, is constructing API-based connections for its B2B integrations. This will help logistics partners integrate into its ERP solutions to request container and shipping details. The company's goal is to create usable mechanisms that unite all elements of the supply chain through modern methods of integration. In this situation, it may be mutually beneficial to encourage an API Alliance model among all parties to increase B2B collaboration and success.

Consulting

For some consumers, APIs are very difficult to understand and integrate with. Though great strides have been made in building world-class Developer Experiences (DX) that aid the API onboarding experience, some complex interfaces warrant extra support. In the IT world, where there is friction there is often a group willing to offer a solution. This may come in the form of consulting.

With a **Consulting** model, a provider provides special assistance to API consumers, helping engineers onboard and develop with their software. Such consulting services go beyond a typical API specialist or developer advocate to offer paid support and maintenance. This may be performed by a dedicated services branch. Companies with open source software, for example, may instead receive a significant portion of their revenue-generating business through custom consulting services. The terms of such arrangements vary greatly.

<i>Taxonomy</i>	External > Partner > Consulting
<i>Type</i>	Direct
<i>Definition</i>	The company provides specialized consulting services to help developer consumers utilize API connections.
<i>Business Model</i>	The company charges for consulting services. Such deals vary greatly.
<i>Developer Consumer</i>	External developer
<i>Key Areas</i>	Consulting departments, IT support service branches
<i>Examples</i>	<ul style="list-style-type: none">• Dwolla: For certain partners requiring special assistance, Dwolla connects them with systems integrator firms to help integrate Dwolla APIs. This helps partners power things like international payments, identity verification, or business verification. Their Partner Ecosystem business model is a referral fee program between them and the system integrator.

Examples (cont'd)

- **Consulting Firms:** Professional service groups such as Deloitte often help their clients integrate with third-party APIs. APIs are embedded into a wide array of digital experiences that such consulting groups develop for their clients. Though these are not their web services, it is evidence that a need is present and many technology companies work in partnership with consultancies to ensure the success of their customers .
- **Acronis:** Acronis Cyber Platform and [Acronis Professional Services](#) provide consulting to help MSPs integrate with their cyber protection APIs.
- **eBay Support:** The [eBay DTS team](#) provides support engineering to their Premium API customers. Support tiers include \$75 per hour (payable in advance), or \$50 by choosing a 4-hour "Jump Start" package for \$250.

Case Study: Acronis

Cyber protection provider Acronis provides software services that protect organizations from lost, stolen, or manipulated data. Their solutions are sold through their IT channels, resellers, and other service providers.

In 2013 Acronis was rearchitected as a platform, built on a set of common APIs across all solutions. These APIs enable services such as backup sync, file disaster recovery, and more to protect customer data. Its 100,000 plus customer base is primarily made of Managed Server Providers (MSPs).

MSPs are typically small, without their own software development teams. Thus, they need outside help. With a unique grasp on the Managed Server Provider (MSP) space, Acronis is positioned to offer complementary professional consulting services alongside its core offerings. Acronis' professional services organization utilizes its APIs to build integrations for its partners. It has built around 30 integrations for customer tickets, remote management, and more to deliver cyber protection.

Take, for example, an organization that oversees a group of dentist practices. Such practices use specialty dental devices and produce unique data sets. The device designers who specialize in producing data-driven dental equipment may seek to differentiate themselves with embedded disaster recovery services. Acronis' professional services branch helps such teams to set up automation and integrations to make that possible.

At the time of writing, Acronis is 17 years old, with 1500 employees 30+ offices worldwide. In 2019, Acronis launched early access for their APIs and SDKs. As of late 2019, Acronis has added the Cyber Cloud Portal as part of its Developer Network.

Standardization

APIs bridge many industries and sectors. As such, data regulations and industry momentum from these unique sectors have sway on the API economy at large. Such influencing bodies may require software vendors to provide APIs due to data rights regulations (which sometimes go beyond the issue of privacy). They may also influence the way they are designed, i.e. to set interoperability standards within an industry.

Thus, our **Standardization** section recognizes the fact that many organizations are compelled to support API initiatives due to standardization by strict regulatory requirements, government influence, or industry consortia. Throughout our observations of these standards, we've identified two unique situations: standards which are **Government-Led** and standards which are **Industry-Led**.

There are many direct and indirect business drivers behind these models:

- **Avoid Penalties:** Government-led industry regulations like the European Commission's PSD2 open banking directive demand programmatic access to data. Meeting regulation thus avoids hefty fines that negatively impact the business.
- **Mutual Benefits:** With standardized integrations across an industry, all boats float higher. So, what's good for the industry is good for business.
- **Customer Experience:** Industry interoperability standards streamline service-to-service communication, improving the customer experience.
- **"Change Phone Plan":** Adopting industry standards makes it easier to onboard new customers to your business who already utilize an integration with the same standard. This is similar to how easy it is to switch cellular networks that share the same protocols.
- **Fear of Missing Out:** Not doing so could be an inhibitor to current and future growth, or other reasons may beckon standardization.

Government-Led

Within certain business sectors — especially ones that are regulated by government — requiring API interoperability methods can make both economic and ethical sense for consumers. Thus, there is mounting pressure on governing bodies to establish and uphold the rights that users have to their data. **Government-Led** API standards can be separated into two groups: (1) **Regulatory**, where the API is a strict mandate and fine is levied against organizations that don't comply, and (2) **Influence**, in which the standard becomes an advantageous commodity that a governing body evangelizes and supports, yet is not enforced by law.

Regulatory

A company may offer an API out of **Regulatory** pressure. Regulatory APIs are prominent in international banking, where PSD2 (Payment Services Directive) has set a stern mandate for API-driven data access. PSD2 legislation encompasses all banks in EU member states and has dramatically ushered a wave of open banking APIs into the financial services world. Government departments themselves may be required by law to provide open data. Other sectors, such as healthcare and government data have set similar guidelines but from one country to the next, compliance may not be mandatory. For Regulatory APIs, the business driver is quite clear: to avoid injunctions and financial penalties.

<i>Taxonomy</i>	External > Partner > Revenue Share > Affiliate
<i>Type</i>	Direct
<i>Definition</i>	APIs are the result of government regulation which enforces data sharing minimum requirements. This is very formal, as companies are penalized if they do not comply.
<i>Business Model</i>	Business creates APIs to avoid hefty fines. Non-regulated institutions in similar industries or other regions may jump on board to stay afloat amid market pressures.
<i>Developer Consumer</i>	Fintech, open data consumers, end-users
<i>Key Areas</i>	Open banking, Fintech, healthcare, government, data control, and privacy
<i>Examples</i>	<ul style="list-style-type: none">• ING Bank: Affected by PSD2 regulation, ING Bank offers APIs to specific FinTech partners to open access to things like user data, account information, bank transfers, and other actions.

Examples (cont'd)

- **Barclays:** Barclays goes beyond the regulation minimum to offer banking architecture as a service. The [Barclays API Exchange](#) provides regulatory services, and also monetizes additional bank microservices as products.

Others organizations that are complying with standards due to regulatory pressure:

[Nordea](#)

[Danske Bank](#)

[GSA](#)

[Department of Veterans Affairs \(VA\)](#)

Case Study: Nordea

With 11 million private customers and 700,000 active corporate customers, Nordea is the largest financial group in the Scandinavian region, headquartered in Helsinki, Finland. Affected by the EU's PSD2 regulation, Nordea was compelled to begin an open banking program.

PSD2 came as somewhat of a shock to the financial world. It necessitated a rapid course change for large financial institutions. Nordea, with a history dating back to the 1800s, is mammoth and typically slow to change. Nonetheless, seeing the API economy growth, Nordea open banking leaders wanted to address compliance and possibly open up new business opportunities as well.

In 2017, Nordea published a beta sign-up page to the Web and promoted it to various FinTech online message boards. They only expected to get around 50 signups. However, to their shock and amazement, within 72 hours they had 300 signups. And, by the end, signups totaled 700. These groups included Fintech and external developers that wanted to experiment with Nordea's APIs.

This enormous interest convinced Nordea to invest more heavily in its external developer program. Thus Nordea designed new services, purchased an API management platform, and improved developer experience with sandboxes. In addition to launching baseline PSD2 compliant APIs to return account information, initiate transfers, confirm payments, etc., Nordea now offers premium services such as an Instant Reporting API, an FX Market Order API, and others.

Nordea was early to market, working closely with FinTechs and actively participating in API community discussions to glean best practices. Through these and other initiatives, Nordea not only avoided PSD2 penalties, but built new revenue-

generating activities to gain a competitive foothold within the burgeoning financial cloud services market.

“Open banking is a movement for taking financial services to the next level of digitalization,” said Gunnar Berger, Head of Open Banking at Nordea. “We strongly believe in co-creating with FinTechs.”

Influence

Some API standards are set by the government but not enforced through injunctions or monetary penalties. Rather, the government may privilege certain technology types as “suggested use,” offering benefits for adoption that are hard to ignore. For the sake of our classification system, we call this model **Influence**.

Within an Influence scenario, the government may also sponsor the creation of standards by industry. For example, this was done with the Office of the National Coordinator (ONC) of the Department of Health and Human Services when it came to the HL7 and FHIR standards for the interoperability of healthcare data between dissimilar Electronic Health Record (EHR) and Electronic Medical Record (EMR) systems. The government did not prescribe a standard specification for the APIs but rather asked the industry members to work those standards out among themselves.

The business effects of Influence are less direct than in the Regulatory scenario. A company may enjoy interoperability benefits, such as integrating with government-supplied datasets or with industry partners.

<i>Taxonomy</i>	External > Standardization > Government Led > Influence
<i>Type</i>	Indirect
<i>Definition</i>	API program is sponsored in part by government influence. Such pressure is not mandated through monetary penalties but the sponsoring governing body plays an important role in offsetting the expenses normally associated with organizing and facilitating across an industry.
<i>Business Model</i>	The software provider may reap benefits from cooperation with government-sponsored initiatives. They benefit from interoperability within the industry to the extent that other peer organizations decide to comply as well.

<i>Developer Consumer</i>	Third-party developers, Industry-specific application developers
<i>Key Areas</i>	Healthcare, Financial
<i>Examples</i>	<ul style="list-style-type: none"> • Blue Button API: The US Digital Service (USDS), in cooperation with The Centers for Medicare and Medicaid Services (CMS), designed and launched the Blue Button API which enables users of different healthcare applications to connect to their Medicare claims data. This standard API aims to improve the interoperability of Medicare claims data between different systems and to allow patients greater access to their claims information. • Green Button Initiative (GBI): The Obama administration mandated an open data initiative to make all government bodies supply machine-readable data. The federal government has also urged various industries to similarly open up their data, particularly in cases like healthcare and energy, following these standards. The GBI was part of this initiative. The GBI outlines a programmatic approach to the sharing of energy usage data that's kept by the various utilities across the United States. • Data.gov: An executive order issued by Barack Obama in 2013 has opened broader effort across other agencies. It is the policy for certain government data to be opened up to the public. Groups like 18F (a part of the US General Services Administration) and the US Digital Service help as a consulting arm to effect this. • San Francisco City: The City of San Francisco provides open APIs as part of DataSF, a smart city initiative. Data types include film locations, building inspection violations, crime maps, and public bicycle parking. US cities such as Seattle, Philadelphia, and others have similar programs.

Industry-Led

The road to API standardization is not always paved by the government. There are many instances of API standardization arising from within industries. In some cases, industries or sectors may come together to create API standards as part of a new or existing consortium. Thus, our **Industry-Led** API business model recognizes the business benefits of supporting industry-specific APIs that were brought about by industry-wide agreement and collaboration.

Industry-Led scenarios may respond to market pressure, technical gaps, or the need for shared protocols. SWIFT, for example, delineated an open standard API for pre-authorization of funds, which could benefit digital banking. Another example is the SCIM API, an open standard that many groups now use for user provisioning.

<i>Taxonomy</i>	External > Standardization > Industry-Led
<i>Type</i>	Indirect
<i>Definition</i>	Industry comes together to create API standards as part of a consortium. Software providers embrace APIs due to industry momentum.
<i>Business Model</i>	API provider sees a mutual benefit across the industry. Interoperability standards improve the customer experience. Standards ease the ability for acquiring new users. Bandwagoning means Fear of Missing Out (FOMO) is avoided.
<i>Developer Consumer</i>	Third-party developers, Industry-specific application developers
<i>Key Areas</i>	Open banking, Fintech, healthcare, government
<i>Examples</i>	<ul style="list-style-type: none">• HL7 FHIR: Health Level Seven (HL7) is an international working group that sets data standards for the healthcare industry. The working group has developed Fast Healthcare Interoperability Resources (FHIR), a specification for healthcare data sharing. FHIR standardized patient Electronic Health Records (EHR) with RESTful APIs. The American Medical Informatics Association endorses FHIR, a consortium of EHR vendors.

Examples (cont'd)

- **SCIM:** System for Cross-domain Identity Management ([SCIM](#)) is an open API standard for managing identities. Salesforce, Gluu, Okta, IBM, and many others implement this in production. SCIM 1.0 and 1.1 were developed under the Open Web Foundation. SCIM is now standardized by the Internet Engineering Task Force (IETF).

- **Durable Data API:** The Financial Services Information Sharing and Analysis Center ([FS-ISAC](#)) is an industry consortium that oversees the [Durable Data API specification](#), a PSD2-equivalent for the United States.

The **TM Forum** is an industry consortium representing over 850 Communication Service Providers and telecoms. In the best interests of making all boats float higher (it's membership), the consortium has led the design and development of [over 50 industry-specific open APIs](#).

Others examples of industry sponsored API standardization efforts:

[SWIFT](#)

[Financial Data Exchange \(FDX\) API Specification](#)

[Cloud Security Open API Working Group](#)

[Mobile Data Plan Sharing](#) (chaperoned by Google)

[Open Geospatial Consortium](#)

Productized

APIs become products when they are commodified and directly monetized. The idea of an API-as-a-Product has become a commonplace approach, offering a programmatic doorway to monetize functionality and data in a way that unlocks a consistent revenue stream for the API provider. Compared with the Partner or Standardization categories, Productized APIs are more self-service and usually public-facing. Often, documentation for these services is open to public view as one way to attract developers. Productized APIs fall into two main subcategories: **Upsell**, where the API is part of a platform subscription tier, and **Coin-Operated**, where the API is directly monetized as a standalone product.

Upsell

An Upsell API business model is when the API is part of a larger product offering within a digital platform. API access is included at a higher subscription tier, or as an add on. The API is a useful element for customers of the platform to increase their customization options. It may facilitate the use and distribution of content between cloud products and/or is shipped with locally installable instances of the platform or solution. An Upsell API can act as an incentive for existing or potential customers to adopt a higher service tier, in essence increasing potential revenue through both the costlier tier as well as through longer term customer retention since organizations are loathe to undo programmatic integrations. In many cases, better platforms or APIs may come along but the organization has a low tolerance for unraveling an existing integration and reprogramming a new one given limited resources and time.

<i>Taxonomy</i>	External > Productized > Upsell
<i>Type</i>	Direct
<i>Definition</i>	The API is part of the product offerings, but only available at a premium subscription plan tier, or add on. The API offers additional customization abilities.
<i>Business Model</i>	API access lures users to a higher fee. It also provides a competitive edge with additional customization options.
<i>Developer Consumer</i>	Clients that use cloud products, Developer that is already a platform subscriber that seeks access to more granular control
<i>Key Areas</i>	Infrastructure-as-a-service, CRM, enterprise services, SaaS platforms with many services
<i>Examples</i>	<ul style="list-style-type: none">• IEX Cloud: IEX Cloud provides financial data. API access to Core Data is included in all plans. However higher tiers unlock Premium Data and add-ons such as the Rules Engine and Cloud Cache

Examples (cont'd)

- **Contentful:** The [Contentful Developer portal](#) offers a suite of RESTful APIs. However, the Contentful [GraphQL Content API](#) is only available within the Enterprise-Grade plan.

- **Salesforce:** There are four editions of the [Salesforce Sales Cloud](#); Essentials, Professional, Enterprise, and Unlimited. Each is available to customers on a different per-seat cost (\$25, \$75, \$150, and \$300, respectively). But in order to gain access to the Sales Cloud's suite of APIs, the organization must subscribe at the Enterprise level. For example, when a customer must move from the Professional tier to the Enterprise tier in order to gain access to the [Salesforce APIs](#), the total revenue to Salesforce from that customer doubles.

Other examples of API providers that monetize their APIs by upselling customers to a more expensive or functional tier include:

Foursquare: Foursquare offers Premium API Endpoints only available for use upon upsell to a premium account tier.

HelloSign: HelloSign upsells with increasing Premium API plans.

Case Study: IQVIA

IQVIA, a large contract research organization, and consultant in the healthcare industry embraces an Upsell model for its APIs. For IQVIA partners, APIs enable cross-product integrations. At IQVIA, moving to an open architecture was a big cultural shift. However, the need for agility was also a driving factor forcing entry into the API economy.

Within the healthcare industry, IQVIA has historically offered custom solutions for its customers in clinical trial research. APIs, however, are now helping IQVIA digitally transform. APIs help productize their integration of flows, allowing them to move away from a provider of bespoke solutions and more towards a turnkey product company. Essentially, it is moving from a custom services and delivery to custom solutions delivery. Here, reusability is key. As IQVIA IT architect Omer Tosun told *ProgrammableWeb*, "we'd rather invest resources in things that would benefit multiple clients and do it right."

APIs at IQVIA utilize LEXI, a data modeling lexicon developed internally at the company. LEXI-fueled services enable a more orchestrated customer engagement

pillar across many areas, providing a reusable platform for healthcare organizations, nurses, and practitioners. IQVIA'S LEXI APIs support services for information management, reporting, social media interaction, events tracking, content management, file review, and compliance.

The company's API model adopts a "land and expand" approach. IQVIA allows consumers to integrate systems within IQVIA's ecosystem for free. Then it charges an additional fee for API calls to external systems.

Coin-Operated

When you think of web APIs of the sort that drive the API economy, you're most likely thinking of Coin-Operated APIs. These self-service APIs are products in their own right. For such programs to be successful, they require a high amount of evangelism, frictionless developer onboarding, a high-quality user experience, comprehensive and easily accessible documentation, API usage dashboards, and other traits; essentially, all the hallmarks of an online SaaS product.

Coin-Operated APIs use a mixture of attributes to price the service. Often, providers utilize a freemium plan to encourage use. Next, tiered models charge per additional use, offering volume discounts, charging a percentage per transaction, charging differently for device type, or other monetization tactics. We've outlined a mix of attributes below that businesses may utilize to optimize revenue generation.

<i>Taxonomy</i>	External > Productized > Coin-Operated
<i>Type</i>	Direct
<i>Definition</i>	Coin-Operated APIs are productized APIs that solve some sort of business function. For developers or businesses that need that function, these APIs make for a great, services-oriented outsourcing option that is often billed on a volume-consumed basis. They are a common form of public API products.
<i>Business Model</i>	Coin-Operated APIs are directly monetized. These programs may use a combination of attributes to optimize their revenue. Most offer free basic accounts until a certain threshold is reached, upon which API calls are monetized. See the Monetization Attributes table below for a comprehensive list of the various coin-operated monetization models in use today.

<i>Developer Consumer</i>	Developers seeking plug and play capabilities, Business leaders that do not want to consume valuable internal resources to reinvent a wheel that offers no discernable competitive advantage over what's offered publicly.
<i>Key Areas</i>	Software-as-a-Service, Business Operational Outsourcing, Geolocation, Cognitive services, AI, Weather, Telco
<i>Examples</i>	<ul style="list-style-type: none"> • Geolocation: APIs such as Google Maps, TomTom, HERE, or Mapbox productize their geolocation services, populating client applications with directions, real-time traffic information and contextual overlay data (eg: Hotels, Restaurants, Hospitals, etc). • Payments: Stripe, Square, Paypal, and others provide APIs to process payments. They monetize by charging for each call and/or a percentage of each transaction value. Such APIs are often integrated into scenarios like eCommerce stores or point-of-sale devices. • Weather: Dark Sky, AccuWeather, WillyWeather, and many more vendors provide paid APIs that programmatically return current weather statistics, helping client applications keep up to date on weather information. <p>Other examples of API providers that offer coin-operated APIs include:</p> <p>Email: SendGrid, SendInBlue, Postmark</p> <p>Telephony: Twilio, Vonage (Nexmo), AT&T</p> <p>Artificial Intelligence: Kairos Facial Recognition, Microsoft Cognitive Services, IBM Watson, Google Translate</p>

Key Monetization Attributes

To optimize revenue for a Coin-Operated API, a provider must scrutinize the value that its API delivers. It may use many attributes to optimize service prices. Think of these attributes as tools in a toolbox for pricing an API.

<i>Per Request</i>	The API has a flat price per call. For example, the API charges \$0.002 per call.
<i>Pay As You Go</i>	A common phrase in API monetization, implying the service offers flexible tiered pricing that will scale to the volume developers require. Often accompanied by freemium and volume-based discounts.
<i>Volume-Based</i>	The price per bundle of API calls is variable on how many requests have been made within a certain time frame. For example, the API charges \$20 for 5,000 requests per month but the fee may be higher if the developer requires 5,001 to 10,000 per month.
<i>Tiered Discounts</i>	Many API providers adopt varying levels of subscription plans. These plans offer bulk discounts when certain thresholds are realized, and/or allow advanced functions. For example, a Beginner plan charges \$20 for 5,000 requests per month. A Professional plan charges \$40 for 15,000 requests per month, offering 3x the bandwidth for only twice the cost. An Enterprise level may charge \$100 for higher or unlimited thresholds.
<i>Enhanced Support</i>	APIs may offer enhanced support features at an additional cost. This could include around-the-clock 24/7 support, direct access to a specific support engineer, or prioritization of certain issues.
<i>Freemium</i>	APIs often have a free tier (freemium) which allows developer consumers to test the API. The end goal is to attract users, create stickiness, and convert to paid tiers. Freemium APIs often have the sleekest, quick, self-service onboarding process.

<i>Per Method</i>	The API assigns varying costs to different methods to reflect different commercial values for objects or resources that are available through the API. For example, an AI service charges \$0.0015 per request to a sentiment analysis method but charges \$0.02 per request to an image detection algorithm since it is more resource-intensive and requires more processing power to compute.
<i>Device Type</i>	The API is priced differently depending on the type of client device. For example, a telephony service may charge varying rates for API calls made from iOS, Android, Desktop, Web Application, or IoT devices.
<i>Rate Limiting</i>	This addresses how many requests can be sent within an allotted time frame. This is different from the volume-basis in that rate-limiting typically refers to smaller time frames, like per second or hour. For example, an API account may have a 10-requests-per-second rate limit. Rate limits must be in place to avoid DDoS attacks that overpower servers.
<i>Pagination</i>	This sets a threshold for how much content can be sent with a single request. At times, the package size may be monetized according to paginated values.
<i>Transaction Fee</i>	The API includes a fee proportionate to the payment processed. For example, Square's starting fee is 2.5% + 10¢ for in-person transactions.
<i>Public vs Private</i>	An API may price requests differently based on the app's intended use case. For example, take the TomTom API monetization program. Compared to its Pay-As-You-Go Public rates, the TomTom API monetization program charges significantly higher rates for its Pay-As-You-Go Private plan, defined as "for any application which is used for internal business purposes only, including asset management."

<i>SLA</i>	API products must also seriously consider what sort of uptime, reliability, and performance to include within their service contract. For example, is a consumer entitled to benefits if endpoints fail without warning, or if the service falls under 95% uptime for a period of time? SLA driven APIs may also provide the customer with a specialized non-public endpoint backed by a higher-grade infrastructure for development purposes.
<i>Geography</i>	Datacenter locations may also determine API pricing. Some services provide data center flexibility, in which consumers are able to choose geographically appropriate data centers through which API calls are serviced. For example, users located in Japan may be serviced by a datacenter in JAPAC while users located in The Netherlands are serviced by an EMEA-based datacenter. This might be structured as a tier, in which higher tiers offer flexibility, and lower tiers do not provide a choice.

Case Study: TomTom

TomTom, a provider of geolocation services, is a model Coin-Operated API. [The TomTom API platform](#) powers services like Apple Maps, the Uber driver app, Azure maps, and many others. One popular feature is its real-time [traffic](#) and [routing](#) features which utilize data aggregated from over 600 million devices.

TomTom's [API pricing model](#) adopts a Freemium tier for new developers, allowing up to 2500 free calls per month. This Pay As You Go model provides free maps and traffic visualization through its SDKs for iOS and Android.

At the time of publication of this report (pricing is always changing), when clients breach the free tier, TomTom offers Volume-Based discounts at 1K calls per month. For example, a 50,000 monthly call limit for public use cases is \$25.00, equating to \$0.50 per call. A 500,000 limit is \$229.00, equalling \$0.46 per call, and so on. TomTom also prices calls differently based on the intended use case. Its prices for "private use" are significantly higher than public use. For example, the TomTom 500,000 monthly API call limit for private use is \$1,719.00, or \$3.44 per call.

The TomTom API program adopts developer inclusion and self-service as guiding lights. Its top tier focus is to be as developer-friendly as possible with public-facing user guides. Quick onboarding and Pay-As-You-Go with Freemium tiers can be a helpful combination for other Coin-Operated APIs.

Marketing

Some API business models generate an ROI through indirect means. In fact, many large API programs are supported purely for marketing purposes. For example, **Packaged** APIs are not monetized directly yet help market the core platform, promoting its sale, adoption, and use. **Audience** APIs help grow a network. In situations like this, opening up an API for general use can encourage additional traffic. With a **Brand Loyalty** model, API programs can spread a brand name and image, bringing advertising benefits. Lastly, **Open API** programs can help a company honor commitments to transparency, open-source, and open doors to the crowdsourcing of data that are shared back through the API while also helping generate leads for the business. These programs are more public-facing than Internal or Partner programs.

Packaged

A Packaged model is when an API is a component of a larger platform, and API access is packaged within the platform's standard access tier. With Packaged APIs, the inclusion of the API entices platform adoption. Since the API is not utilized as an upsell mechanism like the Upsell productized model, this model is more indirect, helping with platform or solution marketing.

Packaged APIs may also be an expected asset for IT offerings, necessary for the platform to operate as a whole. Such is the case for CRM or cloud hosting platforms. It might be that the APIs are shipped with local instances. In these cases, the API is not the focal point of the main product but is a side offering that increases platform value, putting it ahead of competitive offerings that lack APIs and on par with those that do.

<i>Taxonomy</i>	External > Marketing > Packaged
<i>Type</i>	Indirect
<i>Definition</i>	The API is a feature, packaged within a larger platform or solution. Thus, it is part of a larger platform business model.

<i>Business Model</i>	The API is not directly monetized but it is used to market the overall platform. This drives additional revenue in that the product is more capable and customizable. This value-added bonus aids customer retention rates. This is because APIs are “sticky” and, typically, once customers go to the effort of coding integrations with APIs, they are generally loath to replace those solutions.
<i>Developer Consumer</i>	Users who subscribe to the greater platform.
<i>Key Areas</i>	Cloud services, Infrastructure-as-a-service, CRM, enterprise services, hosting
<i>Examples</i>	<ul style="list-style-type: none"> • Domo: Domo is a platform enabling users to “connect and normalize data from any source.” Its offering includes a suite of integration capabilities that involve many API connections. • Sugar CRM: Sugar CRM provides a REST API packaged into its platform offering. Users can download and run SugarCRM locally or host their own instance in the cloud. • Cloud Computing: Many cloud computing platforms provide packaged APIs as part of their core service offerings such as AWS S3 REST API, Amazon EC2, Microsoft Azure Cloud Services REST APIs, or Google Cloud Platform APIs. These APIs represent a way to communicate with computing resources. • Form.io: Form.io is a solution that simplifies the inclusion of data driven forms into progressive web applications (PWAs) while tying those forms to an API-enabled data management back-end. In other words, data that’s collected through the PWA forms can be automatically exposed to other applications through APIs that are bundled into the backend. While the form/ PWA functionality alone probably attracts a fair amount of interest, the API capability that’s packaged with the backend will attract a certain class of customer that might not otherwise be interested.

Audience

A software provider may utilize an API to build an audience or network. These services want as many people as possible to engage with the provider's platform. Another way to phrase this is: monetize the audience, not the features. Syndicating content by way of free public APIs is also called "loosely controlled usage."

With an **Audience** model, a provider offers API access to allow consumers to both connect with the network and contribute to the network (eg: adding listings to Airbnb). Platforms that require extensive databases can extend their network and reach utilizing APIs, enabling programmatic content creation. Listing services, for example, benefit from providing programmatic API access to their partners. The main goal is to increase the amount of content and data within the system and increase usability with a programmatic interface. The end-customers thus perceive the best inventory, and for these platforms, the more customers the better.

The Audience model is different from Partner Affiliate programs in that they do not necessarily offer Revenue Sharing. Such partner programs are unique from Public API programs because they are limited to select partners.

<i>Taxonomy</i>	External > Marketing > Audience
<i>Type</i>	Indirect
<i>Definition</i>	API is utilized as a marketing tool to help engage with an audience or grow a network.
<i>Business Model</i>	The audience, along with community-generated data, is monetized through advertising or reselling user data. Thus, the API is one of many facets of building and growing a following that is crucial for the business to function.
<i>Developer Consumer</i>	ISVs, brokerage groups, Social facing marketers, end-users
<i>Key Areas</i>	Realty, Bookings, Travel, Social media, marketing, news media, listing services, talent, large online communities.

Examples

- **Airbnb Partner API:** The exclusive [Airbnb Partner API](#) is designed for hosts with multiple listings. The API enables programmatic listing to increase Airbnb's database of offerings. This increases interoperability with the platform and increases the number of listings to maximize Airbnb's network effect.
- **Twitter:** Social media networks like Twitter utilize public API programs to grow their base and increase stickiness. For example, the [Twitter Authentication API](#) enables social logins on third-party websites, like Medium. In the early days, Twitter utilized a public developer program to extend their audience to new platforms via third-party applications like Favstar, fm, DailyBooth, TweetDeck, Tweetbot, Echofon and Twittrific. These distributed reach to new platforms and applications.
- **Meetup:** [Meetup](#) provides a free-to-use API for exploring and interacting with the Meetup platform from third-party apps. This could be utilized by event creators to increase their audience base, and in effect, increase overall dependence and engagement with the Meetup platform.

Others:

[TripAdvisor Content API](#)

[Facebook](#)

[Upwork](#)

[Silverfin](#)

[Booking.com Partner Program](#)

[TravelgateX](#)

Brand Loyalty

Many API providers benefit from an increase in **Brand Loyalty**. Thus, a provider may seek to build a brand reputation by externalizing an API. The API helps distribute content to visual external areas, such as client web applications or mobile apps. Since the goal is to increase brand awareness, API development and maintenance may come out of a marketing budget. API products may also utilize a brand loyalty model in concert with other models. For example, the Google Maps API, which invariably results in embedded apps bearing the Google Maps logo, may require specific branding guidelines, even though its core model is a Coin-Operated product.

<i>Taxonomy</i>	External > Marketing > Brand Loyalty
<i>Type</i>	Indirect
<i>Definition</i>	API provider builds brand loyalty by externalizing an API.
<i>Business Model</i>	Brand reputation has a net positive effect, often stimulating additional interest in the platform.
<i>Developer Consumer</i>	External developer
<i>Key Areas</i>	Social media, marketing, news, product listings,
<i>Examples</i>	<ul style="list-style-type: none">• YouTube: YouTube's API Branding Guidelines require that "any YouTube logo used within an application must link back to YouTube content or to a YouTube component of that application." Thus, API usage is always directly benefiting the brand.• Zillow: Though Zillow's API is free, its API Branding Requirements state that "an approved Zillow logo must be displayed directly adjacent to any data from the Zillow API." For Zillow, each API type requires a specific note. Clients that consume Zillow's GetRateSummary API, for example, must include the phrase "See more mortgage rates on Zillow" with an appropriate backlink.

Examples (cont'd)

- **NY Times:** The [NY Times API](#) requires that all applications must be accompanied by a Times API logo on any page or screen that displays Times API content or data. The logo must link directly to "<https://developer.nytimes.com>" (thereby driving additional interest in the media organization's developer program).

Other brand guidelines for API programs:

[Spotify](#)

[Dropbox](#)

[Orange](#)

[Slack](#)

[TokBox](#)

[Walkscore](#)

Open API

Open APIs are public APIs that are freely accessible. They may still have security measures in place but are generally free for public consumption. Open APIs can bring many marketing benefits to a business. For one, an Open API model may increase **Transparency** into a company's operations, helping boost its PR image. An Open API model can help build a **Community** around a company, which increases the engagement, interest, and marketability surrounding a brand. A company may seek to develop Open APIs as part of an open-source initiative — such a **Lead Generation** model can eventually help acquire sales. Lastly, a provider may choose a **Crowdsourcing** approach, in which Open APIs are utilized by the developer community for collaborative development and innovation. All these activities can benefit a company's image and increase perceived value. A commitment to openness can generate stickiness, which in the long-term can be converted to sales.

Transparency

By making data available to the public, companies can hold their business leaders accountable and ensure the company is acting in the best overall interest. User data rights, political transparency, and business ethics are in the spotlight more than ever. With misinformation on the rise, the public demands accountability from tech platforms. According to [Slack](#), “In an increasingly interconnected world, transparency is the new gold standard.” Thus, efforts to publicly open such data can positively affect transparency greatly.

<i>Taxonomy</i>	External > Marketing > Open API > Transparency
<i>Type</i>	Indirect
<i>Definition</i>	APIs are made open to meet transparency initiatives. Otherwise paid services may be discounted or made free for public benefit or humanitarian causes.
<i>Business Model</i>	A transparent API business model increases trust and public perception around an organization and can help avoid corruption. Transparency may also unlock future business or partnership opportunities.
<i>Developer Consumer</i>	ISV, external developers
<i>Key Areas</i>	Advertising, User Data, Inclusion, Business Ethics, Accountability, Anti-corruption, Public Transportation, Government, Reporting
<i>Examples</i>	<ul style="list-style-type: none">• Google Ads: According to a recent blog post, Google Ads will soon roll out new APIs that enable advertising companies to disclose detailed ad information, increasing transparency in digital advertising.• Political Transparency: In 2019 Mozilla led an effort encouraging Facebook to open an API to allow greater transparency into political ads that appear on the platform. An open letter called for “a functional, open Ad Archive API that enables advanced research and development of tools that analyze political ads served to Facebook users in the EU.” While the effects have yet to be seen, it represents a growing effort to increase transparency through programmatic means.

Examples (cont'd)

- **Buffer:** As an organization, Buffer has a high commitment to transparency, even going so far as to publicly post each [employee's salaries and diversity data](#). Such data could be made API-accessible to further a cause for transparency.

Others examples:

Government Open Data Portals such as [Transparency.Treasury.gov](#) by The Bureau of the Fiscal Service.

Since its very early days, Salesforce has been extremely transparent about the availability and status of the systems that power its various clouds. Through [trust.salesforce.com](#), the company not only advertises the status of all of its systems, it makes the data programmatically available through RSS feeds (which are a form of API).

Community

Community is powerful. Internal corporate culture is important, but building a community of external developers around open APIs can increase overall marketing efforts tremendously. This is particularly true in open source scenarios where the developer community isn't just consuming the API provider's APIs but is also developing and submitting code that improves the APIs themselves. Developer communities can be encouraged through incentive programs, meetups, forums, free online content, and other advocacy activities.

<i>Taxonomy</i>	External > Marketing > Open API > Transparency
<i>Type</i>	Indirect
<i>Definition</i>	APIs are made open to meet transparency initiatives. Otherwise paid services may be discounted or made free for public benefit or humanitarian causes.

<i>Business Model</i>	A transparent API business model increases trust and public perception around an organization and can help avoid corruption. Transparency may also unlock future business or partnership opportunities.
<i>Developer Consumer</i>	ISV, external developers
<i>Key Areas</i>	Advertising, User Data, Inclusion, Business Ethics, Accountability, Anti-corruption, Public Transportation, Government, Reporting
<i>Examples</i>	<ul style="list-style-type: none"> • Magento: Magento is one of the few open-source eCommerce solutions. Magento has a strong developer community built around its open-source software and API programs with its Magento Community Forum, events worldwide, etc. This support and passion has led to 250,000 online stores. • TensorFlow: TensorFlow was initially developed by the Google Brain team and is now under an Apache license. It has APIs and SDKs available in several languages both for constructing and executing a TensorFlow graph. It could be argued that supporting open-source tech communities has in part further solidified Google's reputation within the artificial intelligence spectrum. • Tesobe: A community banking perspective to connect community members is a sustainable solution to being digitally transparent and open. One example is the Tesobe Open Banking platform. The Tesobe team holds Open Banking hackathons, training sessions, and publishes thought leadership to help build its brand and stature as a standard bearer of Open Banking. <p>Others examples of community-driven API programs:</p> <p>StackExchange</p> <p>Shopify</p>

Lead Generation

Open APIs and open source projects related to APIs can also act as a Lead Generation mechanism. Lead generation is a business process whereby an organization typically offers some free functionality, download, or other information in exchange for an individual's personal contact information (ie: name, email address, business title, etc.). That information is saved into a database of hot leads (potentially interested parties) that's subsequently used by the organization's sales and marketing teams to prospect for new customers. By encouraging open use of a certain provider's API, that provider is uniquely positioned to accumulate new leads. In the long run, such API programs can convert these leads into customers who pay for ancillary or upgraded services.

<i>Taxonomy</i>	External > Marketing > Open API > Lead Generation
<i>Type</i>	Semi-Direct
<i>Definition</i>	Open APIs help generate awareness for the company. Posting open source projects related to APIs on Github, or packages such as Software Development Kits (SDKs) and other tooling on npmjs.com , pypi.org , etc. can increase usability with the API service and extend reach to new developer users, which can be targeted for revenue-producing opportunities. For example, an API provider could pair this business model with the consulting model in a way that results in actual revenue.
<i>Business Model</i>	Open initiatives increase exposure for a company and act as a marketing funnel. This traction and usage behave as a sales funnel to generate leads and sell customers other products or services.
<i>Developer Consumer</i>	External developer, potential developer users
<i>Key Areas</i>	SaaS side projects, Open-Source

Examples

- **GROQ by Sanity.io:** [Sanity](#) provides modern CMS architecture. [GROQ](#) is Sanity's open-source query language for JSON documents. GROQ can be run on [this page](#), from a [CLI](#), or within Javascript apps. By open-sourcing GROQ, and releasing a web-based playground for queuring any JSON document, Sanity has built open-source public-facing resources with calls to action toward using the commercial Sanity software, thus driving leads.
- **TomTom Helper Libraries:** TomTom offers geolocation as a service through its APIs. TomTom's [Github](#) hosts 60+ repositories, including code examples for their [iOS](#), [Android](#), [Web](#). Other packages include [speedtools](#), [openlr](#), and [commit-checker](#). Such packages can improve developer usability and help generate interest in TomTom's commercial services.
- **OAuth.tools by Curity:** [OAuth.tools](#), built by API security experts [Curity](#), is a valuable public resource for exploring the many types of API authentication flows conforming to the OAuth standard (Implicit Flow, Code Flow, Device Flow, and nine others). This helps API owners discover the nuances of the OAuth protocol for API authorization delegation. Visitors can paste a Javascript Web Token (JWT) and decode its header, payload and signature, and build complex flows that connect to any OAuth server. Users who encounter the tool are naturally lead to explore Curity's other commercial solutions.

Public Good

Some APIs may be part of altruistic social good programs within large organizations or be part of humanitarian efforts at non-profits. Such APIs are typically open in nature.

<i>Taxonomy</i>	External > Marketing > Open API > Public Good
<i>Type</i>	Indirect
<i>Definition</i>	The API is used for greater social good with altruistic purpose. Such APIs open source (and at times crowdsource) data sets.
<i>Business Model</i>	Government organizations, non-profits, and select corporate divisions may be required to provide APIs for social good, to appeal to environmental, sustainable, and humanitarian/social good causes.
<i>Developer Consumer</i>	Citizen developer
<i>Key Areas</i>	Aid, World Health, Natural Disasters, "Data for Good," Sustainability, Climate Change, Environment, Non-Profit
<i>Examples</i>	<ul style="list-style-type: none">• Natural Disasters: According to a Nature.com report, Scientists working under the motto "data for good" are scrutinizing location and activity data donated by some of the world's largest telecommunications companies. By anonymizing phone records of tens of millions of people in low-income countries, aid organizations can track and help support the movements of large populations during natural disasters such as earthquakes, flood, disease, famine, and other crises. API-enabled data, in this case, could increase ubiquity and accessibility tremendously.• Sunlight Foundation: The Sunlight Foundation is a non-profit that "uses civic technologies, open data, policy analysis and journalism to make our government and politics more accountable and transparent to all." The foundation created APIs that enable better transparency into government actions. Now maintained by external groups, these APIs include a Congress API, Capitol Words API, Energy Legislation Tracker, Bill Tracker, and other tools.

Examples (cont'd)

- **GDC:** The National Cancer Institute provides a [Genomic Data Commons \(GDC\) API](#). This allows developers to query and download GDC data while at the same time allowing other developers and organizations to submit data in a way that improves the value and efficacy of GDC's corpus.
 - **AMEE:** Avoiding Mass Extinction Engine (AMEE) Provides [free APIs](#) that return millions of environmental data elements including sustainability data, carbon emissions factors, and models. AMEE's "sustainability at scale" is an easy way for developers to discover, consume, and integrate environmental data.
 - **OpenCorporates:** OpenCorporates provides company data as a service. Its [Public Benefit](#) program provides free API account access for certain public benefit projects.
- Others:
- [CGAP API](#)
 - [TCIA Programmatic Interface](#)

Outliers

This outlier section consists of potential business model types. They may make sense in theory but are not proven enough in practice to warrant a specific category above.

Recruiting

Some API programs are more about recruiting than productization. Companies that host internal or external API hackathons give developers an opportunity to experiment with services to spur innovation. It's not uncommon for recruiting teams to often attend these events to search for talent to acquire.

Alternative Affiliate

Some Affiliate API Programs may not technically payout developers according to revenue share. Advertising agreements that utilize APIs, for example, may pay out in Cost per Click, or pay per impressions.

Directory

Putting your API into a management sphere or gateway can impact the business model. Monetizing your API by hosting it through a third party gateway may denote its own strategy. This is especially true for directory services that aggregate APIs and allow a standardized shim for external developer usage. The benefits of taking this approach include a reduction in time to market for coin-operated APIs because the directory is a destination that's already used by developers to discover APIs. The downsides could include a loss of customization capabilities, hosting and API endpoint options, and fine grain API management and governance capabilities.

Mashups

Companies may use multiple external APIs to assemble a working value proposition. This could come in one of two forms. The first is when a single API is used to abstract a workflow that's driven by a string of other APIs (some or all of which could be third-party). The value and selling proposition then lies in the service composition. The second is when the API provider creates a standard API to access multiple APIs of the same type. The value to the consumer, in this case, is derived from the commoditization effect that the "standard" API introduces to a collection of similar and often competing APIs. For example, there are many facial recognition APIs from different providers. Another API provider could come up with a standard API that offers the developer a single point of access to all of the face recognition APIs. This might be helpful to a law enforcement agency that wants to submit images to multiple recognition engines simultaneously.

On the other hand, for developers that want to reduce the switching costs between recognition engines, a standard API that works with all of them can help to eliminate the development of new source code when a switch is warranted.

The Other Side of the API Economy

How do we account for business models that target the API providers themselves? Many vendors now exist within the API economy itself, offering solutions for API management, Gateways, API testing, API security, Identity and Authentication, API Monitoring, Integration Platforms, Documentation as a Service, and others. However, this segment of the API Economy is out of scope for the business models that we're discussing in this document.

Methodology For API Business Model Creation

Now that we have covered the majority of API business models, how do we put them to practice? In this section, we'll briefly overview some pragmatic steps you can take to locate value, and determine the ingredients of your own API business strategy.

Airlines Example:

Throughout these steps, consider how an airline may seek to utilize APIs within its ecosystem.

1. Find Potential Value

First, audit your internal Mode 1 infrastructure. Evaluate all your systems of record to see what data and functionality could be exposed. What does this data look like? Are there gaps within your technology that could accelerate business?

The airline has valuable passenger data and rewards club member data. It has flight data that could be utilized. However, few perks are being offered to retain customers.

2. Perform Market Research

Next, check the competition in your space or in tangentially related industries. For example, airlines are not just in the airline business. They're in the travel industry. Drugstores aren't just in the pharmacy business. They're also in retail. Who's doing what in your industry now? How are they accelerating digital transformation or productizing, or using other means to generate ROI utilizing their digital assets? How are other companies utilizing APIs including third-party APIs to build game-changing customer experiences? From the customer's point of view, where are the gaps in your customer experience?

Airline researches space to see if other airlines are partnering with rideshare companies. It researches the typical discounts offered by such partnerships in other industries. Airline looks to ancillary transportation methods and notices a customer experience gap. It has an opportunity to participate in shuttling passengers to the airport.

3. Define Unique Value Proposition

Next, define what your comparative value is and how you could digitally transform your organization. Define what infrastructure is required to deliver said value effectively. What will the end value proposition look like?

Airline traditionally defines its role as a transportation coordinator. It must offer improved experiences by integrating with partners to co-create compelling customer experiences that go beyond transportation coordination.

4. Define The Customer Experience Outcomes

Next, define target customer experience outcomes as well as outcomes for the business. Determine how APIs will enable those experiences. Do you have the architecture in place? Can you showcase success stories to encourage internal adoption?

Airline seeks to encourage repeat customers by offering a streamlined easy experience. Customers enjoy the ease of use. The airline benefits by having a more devoted customer base.

5. Establish Partners to Co-Create The Experience

In order to co-create experience, seek out partners to integrate with. What partners do you need to offer this experience? Can you mashup APIs to co-create an experience utilizing many separate core competencies?

Airline partners with Uber to offer discounted airport transportation for travelers. It fully realizes the potential for integrating with other niche services to remedy inefficiencies.

6. Discover Expenses Involved

Next, discover the cost of integrating with partners. Evaluate the cost of exposing data and developing internal APIs. Outline the return in the form of business KPIs needed to ensure the API program is worth it.

Airline creates a budget that incorporates cost estimates from API usage, internal development efforts, integration time and effort, and ongoing maintenance effort.

7. Build The Developer Experience

Now that the end experience is envisioned, and the partners are known, it's time to build out your Mode 2 infrastructure. In this phase; think beyond your core competencies. At this point, the devil is in the details: what's the best end possible customer experience? API products must create self-service guides catered to defined partners.

Airline builds components necessary to architect its vision.

8. Set Monetization Parameters

This last modeling step involves fine-tuning. Set thresholds and tiers for direct monetization models if necessary. Set measurable goals and APIs if the business model leans towards indirect benefits. Adjust to find a middle ground to optimize ROI.

Airline determines correct monetization parameters to offer ROI and retain continual customer loyalty.

Takeaways For API Business Modelling

There is high value in data and functionality. However, more than ever, there is value in how it is accessed. As we've witnessed, APIs are a strong choice to transform a component, business, or even an entire industry. An API strategy can bring Long Term Value (LTV) and committed developers.

- **Complexity:** API business model complexity has increased since John Musser's first coverage in 2013. It should be noted that a company may have varying APIs that serve different functionality, and each API may use different models in tandem.
- **Nothing is Free:** Every API has a model behind it. Even the seemingly "free" and indirect ones. Companies can utilize APIs to extract value down to their bottom line in the form of money saved or new revenue earned without having to be directly transactional.
- **Net accretive vs. cannibalized:** Generating and preserving unique income is essential for net revenue protection. Meaning, new forms of revenue generation should be overall accretive, not cannibalize existing core business models. Thus, API providers must find the optimal balance between externalization and proprietary knowledge.

Brainstorming how these models fit into your organizational strata is a good exercise for any chief executive seeking new revenue streams, CTO looking to innovate, or API product manager assigned to monetizing digital assets. What's most interesting is that the majority of enterprises don't even link transactional models to their APIs. Our experience working with API providers for more than a decade demonstrates that most of them prefer indirect business models.