{"logo":"API Evangelist"}

API Industry Guide

API Management

August 2015

by Kin Lane, the API Evangelist

This is meant to be a field guide to the fast changing world of API management, providing you an overview of companies, tooling, common building blocks, and some of the latest news from across the landscape.

Overview API Management

API management was not born out of any particular group or standard, it was defined by the early API pioneers like SalesForce, Amazon and Ebay, which was then iterated upon by newer players like Twilio, SendGrid, Stripe, and others.

As a standard thing, the first company to offer API management as a service was Mashery in 2006, who helped set a common approach to providing API portals and coined the term "business of APIs".

In 2015 there are now over 40 API management service providers providing a wide range of services from access management and analytics to complete API gateway services to help you deploy and manage hundreds or thousands of APIs from internal resources.

I've always lumped everything about APIs into a bucket called API management. I'm slowly pulling out the separate pieces like design, deployment, monetization and evangelism, leaving API management as the technology, business and politics of operating multiple APIs for your company, or as a business.

There are two main paths that API providers take, either DIY approach where you have the resources and launch your own portal, and cobble together various tools and services to make it all work. The second path is talking to several of the API management providers and discovering which one will best help you with your goals.

Whichever path is taken, the goal of this paper is to provide you with the building blocks and overall knowledge you will need to be successful. I have studied the tools and services offered by the API management service providers, but more importantly I have looked at all the public APIs available today, and tried to reflect the best practices they've established through successful operations.

API management should always begin with a portal area, where you host the operations of one or many API resources, as well as multiple public platforms like Github. API management involves not just the management of the technical API interfaces, but the blog, forums, documentation, support desk and other critical building blocks of API operations.

Any company or individual that is new to the game, I highly recommend spending 2 weeks studying the top API platforms and understanding how they run their own API management operations. Next I recommend you pick 2-3 of the API management service providers and give them a call, have conversations with them about your goals.

API management is not something you read the book on, then execute flawlessly. API management is about ongoing, iterative research and development around your companies data and resources. Your API management strategy will define the next phase of your business development.

Common Building Blocks For API Management

As the API Evangelist, I have spent five years studying how API operations working, a good portion of this work is focused on identifying building blocks, in hopes of establishing, easy to understand modules anyone can pick up, and include as part of any API strategy.

In 2010 I started API Evangelist as a research project to look at how the top 250 popular APIs were doing business. I spent the summer of 2010 looking at these API providers and established a list of building blocks that each provider used to manage their API operations.

I've been maintaining this list for 5 years now, looking at thousands of APIs and assessing the building blocks used by the most successful of API companies. I've tried to standardize these and establish a potential list that any API provider could consider when planning their own API management strategy.

These are the most common examples of API management building blocks, grouped into meaningful areas of operation.

Onboarding

When it comes to APIs, first impressions are everything. One of the common things I see with the most successful APIs out there, is a simple, self-service approach to learning what an API does, and how you actually use it. There are proven approaches to reducing friction for developers when on-boarding with an API, and here are a few of them.

Documentation

Documentation is the number one stumbling point for developers when trying to integrate with an API. Simple, consistent, and increasingly interactive API documentation, is required for developers to get up to speed. If documentation is not clear and consistent, developers will easily go elsewhere. Make sure and consider these common documentation building blocks.

- **Documentation** - Quality API documentation is the gateway to a successful API. API documentation needs to be complete, yet simple--a very difficult balance to achieve. This balance takes work and will take the work of more than one individual

on an API development team to make happen. API documentation can be written by developers of the API, but additional edits should be made by developers who were not responsible for deploying the API. As a developer, it's easy to overlook parameters and other details that developers have made assumptions about.

- **List of Endpoints** - Before a developer is thrown into the full detail of API documentation, it helps to introduce them to all available API endpoints, getting them acquainted with the resources available. A simple listing of all endpoints provides a quick introduction, that will prime developers for a deeper dive. After reviewing all API endpoints a developer can start to imagine how their application will integrate with an API, further understanding the value the API will bring to their application. Sometimes it's hard to see the 100K view of an API from regular documentation, start with just listing the API endpoints.
- **Interactive Documentation** - There is a new movement in API documentation, one that is moving beyond static, often boring documentation and into a new realm where API documentation is live and interactive. Following in the footsteps of API explorers these new interactive documentation formats like Swagger and Mashery I/O Docs, allow developers to authenticate, navigate endpoints and make requests with live responses returned. In a little over a year, interactive API documentation has gone from a new innovation of a select few APIs, to being a standard offering among many of the leading APIs in the space. There is no better way to get your developers acquainted with an API, than allowing them to interact with your API while reading documentation--turning API documentation into a hands on experience.
- **API Explorer** - API explorers allow users to make calls and explore REST APIs using a web interface. The simplicity of REST has contributed to the extreme growth in the number of Web APIs in the last year, and API explorers are going to fuel this growth. API explorers put the power of Web APIs in the hands of non-developers, allowing journalists, students, politicians, and any tech savvy Internet user to access the data and functionality available via APIs.
- **Error Response Codes** - Errors are an inevitable part of API integration, and providing not only a robust set of clear and meaningful API error response codes, but a clear listing of these codes for developers to follow and learn from is essential. API errors are directly related to frustration during developer integration, the more friendlier and meaningful they are, the greater the chance a developer will move forward after encountering an error. Put a lot of consideration into your error responses and the documentation that educates developers.

Authentication

Requiring authentication for API is commonplace, despite many myths that all APIs are public. Part of this reality includes explaining authentication to potential API consumers. This is one of the most important areas we should not be re-inventing the wheel, and follow the common building blocks around API authentication.

- **oAuth** - Providing an oAuth layer to API operations, securing high value APIs, while also opening up a conversation between an API platform, developers, and end-users regarding the access of their content, and valuable data.
- **Basic Auth** - Usage of the basic authentication format that is part of the standard HTTP operations, employing a user's username and password as credentials for accessing API resources.
- **Authentication Overview** - Always provide an overview of what type of authentication is provided for an API. Don’t assume developers will know anything about Basic Auth or oAuth. Walk developers through goals behind authentication, with links to tutorials regarding authentication technology. If an API employs oAuth, make sure and take extra special attention to provide clear instructions on how to use, as well as language specific code as part of your API SDKs. After poor API documentation, oAuth integration is the number one stumbling point for API developers.
- **Key Access** - Providing simple tokens, often called API key is a common way to provide access to APIs. Keys are issued uniquely to each developer and even per application. Developers can reissue keys, and manage their access. API providers often use API keys to track on how developers are using an API through analytics attached to API keys.
- **Authentication Tester** - When possible, provide a testing tool for authentication. From key and Basic Auth to oAuth, allow developers to enter their keys or tokens and validate the credentials they are using, to make sure they are using the proper credentials. A simple tester can provide quick validation that they are doing it right or show them where they are making a mistake, eliminating serious frustration while programming.

Code Management

When i comes to API operations, and integration, there is lots of code to consider, from samples, to working applications from successful integrations. I've identified a number common building blocks the leading API providers are using to successfully manage their code.

- **Github** - Using Github for managing of code samples, libraries, SDKs, and other supporting elements of an API platform is essential to operations, while also providing another channel for a platform to engage with its developers.
- **Application Gallery** - Complete, functioning applications built on an API is the end goal of any API owner. Make sure and showcase all applications that are built on an API using an application showcase or directory. App showcases are a great way to showcase not just applications built by the API owner, but also showcase the successful integrations of ecosystem partners and individual developers. Do not hesitate populating an application showcase with your own active or starter kit applications. As with all of the API code, make sure and provide as liberal licensing as possible to ensure developers can be successful with use.
- **Open Source** - An API is inherently an external part of a company. The documentation, code samples, SDKs, starter kits, platform development kits and any code related to an API, should be considered external intellectual property and licensed accordingly. Consider open sourcing all of the code associated with an API. Open source will fuel the innovation that is already present in API ecosystems, further reducing the friction experienced by developers in successfully integrating their applications and businesses with an API.
- **Starter Projects** - Many API owners are going beyond just code samples and generic SDKs for their API ecosystems and providing open-source, private label applications built on top of an API that developers can download, modify and deploy. These projects go by many names, but are commonly known as starter kits or projects. Starter kits can act as code samples, and may contain a version of an SDK, but provide a complete application that reflects common integrations with an API. As with samples and SDKs, start kits will speed up integrations, providing developers with the path of least resistance from registration to active API integration.
- **Community Supported Libraries** - In addition to providing your own company developed libraries, showcasing the libraries of trusted developers fro within the API community.

- **Code Builder** - A tool allowing for the generation of client code for various API endpoint.
- **Code Page** - A page dedicated to providing access to code resources, whether samples, libraries, SDKs, PDKs, or starter projects.
- **SDKs.io** - Making sure your profile on SDKs.io is complete, with up to date SDKs available so users can use in their integration.

Code Libraries

One of the quickest ways to make an API speak someone's language, is speak the language they are most familiar with. The leading APIs always provide simple code libraries, in a wide variety of programming languages.

- **JavaScript Library** - Code libraries in JavaScript for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.
- **PHP Library** - Code libraries in PHP for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.
- **Python Library** - Code libraries in Python for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.
- **Ruby Library** - Code libraries in Ruby for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.
- **Node.js Library** - Code libraries in Node.js for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.
- **Java Library** - Code libraries in Java for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.
- **.NET Library** - Code libraries in .NET for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.
- **Scala Library** - Code libraries in Scala for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.

- **Haskell LIbrary** - Code libraries in Haskell for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.
- **ColdFusion Library** - Code libraries in ColdFusion for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.
- **Perl LIbrary** - Code libraries in Perl for developers to use in their native language. Libraries are simple libraries meant to introduce developers to any API and are usually not meant for production environments.

Software Development Kits (SDK)

Going beyond just samples, that assist API consumers in on-boarding with an API, an increasing number of APIs are also including full blown software development kits (SDK) that provide comprehensive coverage of an API, including authentication. Make sure and provide as wide of a variety as you can, when delivering API SDKs.

- **JavaScript SDK** - A client side JavaScript SDK for integrating with one or many APIs.
- **Node.js SDK** - Software Development Kits (SDK) in Node.js for developers to use when integrating with their applications in the language they are familiar with. SDKs should be expected to be used in production environments.
- **Python SDK** - Software Development Kits (SDK) in Python for developers to use when integrating with their applications in the language they are familiar with. SDKs should be expected to be used in production environments.
- **.NET SDK** - Software Development Kits (SDK) in .NET (VB or C#) for developers to use when integrating with their applications in the language they are familiar with. SDKs should be expected to be used in production environments.
- **PHP SDK** - Software Development Kits (SDK) in PHP for developers to use when integrating with their applications in the language they are familiar with. SDKs should be expected to be used in production environments.
- **Ruby SDK** - Software Development Kits (SDK) in Ruby for developers to use when integrating with their applications in the language they are familiar with. SDKs should be expected to be used in production environments.

- **Java SDK** - Software Development Kits (SDK) in Java for developers to use when integrating with their applications in the language they are familiar with. SDKs should be expected to be used in production environments.

Mobile Management

Mobile devices are why the API industry is even a thing. It wasn't until we needed to deliver valuable digital assets to mobile devices, that the true potential of APIs became clear. There are a number of common building blocks to consider as when managing the mobile side of API operations.

- **Mobile Overview** - Provide an overview page, dedicated to mobile application development. Page should give equal time to each platform, including iOS, Android, and Windows.
- **iOS SDK** - The trends is clear. Apple is the dominant platform for mobile application development. API owners need to have a clear understanding of what iOS developers are needing for both iPhone and iPad application development. When possible, provide iOS specific code samples, SDKs and other resources iOS developers can employ to make their API integrations successful.
- **Android SDK** - When it comes to mobile development, Google's Android platform is definitely the number two player in the space, and warrants similar attention as the iOS building block. Consider providing Java code samples and SDKs specifically for the Android mobile platform. Android is picking up momentum in the space and with new devices being released all the time, API owners can't ignore the platform as a serious contender.
- **Windows Mobile SDK** - Providing mobile focused SDKs for developers to build Windows mobile applications.
- **HTML5** - While the native app vs. HTML5 app development battle rages on, API owners have to closely pay attention to HTML5 as a viable alternative offering for their API developers, right alongside iOS and Android. For many web developers, HTML5 is a natural transition to mobile development--a factor that may tip the future toward more HTML5 mobile implementations. Big players like Apple, Facebook and Google are investing heavily in the future being HTML5, which sends the signal to API owners, that they should do the same.
- **Appery.io** - A new breed of mobile development platforms are emerging, and the leader of these is a product called Appery.io, from established technology company Exadel. Using Appery.io developers and even non-developers can build

cross-platform mobile applications using a GUI building environment. Appery.io provides a suite of API connectors allowing for rapid mobile application development using APIs, with the ability to then deploy as native iOS and Android as well as web mobile applications. API owners should consider working with Exadel to deploy an API connector for their companies API.

Platform Development Kits (PDK)

Software Development Kits build bridges between an API and specific system integrations, in a specific language. Platform Development Kits build bridges between an API and common platforms that operate in the cloud--using APIs. There are some common platform integration elements I'm seeing emerge across the leading API providers.

- **Wordpress** - Providing ready to go WordPress integration, allowing developers, and sometimes even non-developers to immediately put an API to use through their WordPress site(s).
- **Heroku** - Providing ready to go Heroku integration, allowing developers, and sometimes even non-developers to immediately put an API to use through their Heroku platform account.
- **Drupal** - Providing ready to go Drupal integration, allowing developers, and sometimes even non-developers to immediately put an API to use through their Drupal site(s).
- **SalesForce** - Providing ready to go SalesForce integration, allowing developers, and sometimes even non-developers to immediately put an API to use through their SalesForce account.
- **Joomla** - Providing ready to go Joomla integration, allowing developers, and sometimes even non-developers to immediately put an API to use through their Joomla account.
- **Google App Engine** - Providing ready to go Google App Engine integration, allowing developers, and sometimes even non-developers to immediately put an API to use through the Google Cloud platform.
- **Chrome Extension** - Providing ready to go Google Chrome browser integration, allowing developers, and sometimes even non-developers to immediately put an API to use in their browser, with a Google Chrome extension.

- **Firefox Add-On** - Providing ready to go Mozilla Firefox browser integration, allowing developers, and sometimes even non-developers to immediately put an API to use in their browser, with a Mozilla Firefox add-on.

Self-Service Support

APIs are a 24/7 type of operation. You are doing business around the globe, and there should always be self-service support options. As with the on-boarding process, self-service is critical for developers to integrate on their schedule, get access to the resources they need, as well as the help they need, exactly when they need it. There are a host of self-service support options that have emerged, that should be evaluated as part of API operations.

- **Forum** - Forums have become an essential building block in API communities for self-service support. A well moderated, active forum can evolve an APIs development area into an actual community. All forum communities will require the API owner to engage developers, keeping conversations active and questions answered, but with the right developers your forum can become self regulating--with opportunities for more senior developers to answer the questions of newer, more junior users, providing potentially free resources for an API owner.
- **Forum RSS** - Going beyond just a forum, and providing an RSS feed for all forum activity. Something I don't see with all API providers, or forum platforms, but is valuable as a signal when I do see one.
- **Knowledgebase** - Providing a single repository of content, organized by title, and tag, allowing you to search for keywords and phrases, as well as browse for what you need. All knowledge about an API is curated within the knowledgebase, and updated over time.

Direct Support

Beyond self-service, their should be easy to find paths to receive direct support, allowing API consumers to get free, or even paid support to the problems they are facing. These are the common building blocks I include when thinking through direct support strategies for API operations.

- **Ticket System** - Providing developers with a support ticketing tool, using a custom system, or via a popular platform like Zendesk, can be the healthy way to support the needs of an API ecosystem. Not all developers will want to publicly post their problems, and support tickets can be a very organized way to handle the direct

support needs of developers, allowing API owners to respond quickly to easy questions, but also allowing them to organize larger scope items into lists that can be used in API product development, providing a direct link between developer support and the API roadmap.

- **Email** - An email is a pretty easy way to provide support to a developer ecosystem. If users see an email, they will know that they can get the help they will need when integrating an API.
- **Paid Support Plans** - Going being just direct, or indirect support and providing paid support tiers for developers to use, allowing them to pay by the minute, hour, or project, and get the intimate, direct support they desire.
- **Contact Form** - A contact form is a pretty proven way to allow API consumers to submit a comment, message, or request for help. Users are generally used to using a contact form, and will easily identify it as a place they can go for help.
- **Phone** - Much like email, a phone number can be a solid choice for API developer support. Especially if you have a dedicated, partner target audience. Phone can be the instant gratification that developers need when they hit problems, get their questions answered and move forward with their API integration. Not all API providers will have the resources for phone support, but in some circumstances it will do the trick.
- **Office Hours** - Open office hours is a great way to provide direct support for developers, in a controlled and sustainable format for API owners. Many popular APIs post office hours each week, giving developers an open time they can engage with support representatives via Skype, Google Hangouts or sometimes even in person. Consider the possibility of using API open office hours to support an API community.
- **Calendar** - An active API will have many events that can be shared with its community, ranging from conferences, hackathons and meetups the API provider will be attending, to industry related events that developers can benefit from. A published calendar is a great way to publicize these events, while also showing that the API is actively engaged within the API community and beyond.

Communications

Most challenges in API operations can be overcome through a healthy dose of communication. Regularly pushing out communications around platform operations, and the overall road-map keep, developers informed, and problems to a minimum. Making communications a two-way

street, goes even further to reduce issues. Here are some of the communication building blocks I recommend.

- **Blog** - An active blog can provide a quality SEO presence for an API, attracting developers and businesses to the API. Secondarily a blog can provide essentials communications for the developer community. While researching this white paper and reviewing 6000+ APIs, a blog is the number one way I could tell when an API is dead and nobody is supporting the community. A blog can easily provide the communication to keep an API active and growing, while also be the barometer of whether developers should steer clear of an API.
- **Blog RSS Feed** - Allowing for users to consume a blog in feed readers, and other sites or applications is critical to getting the necessary reach for any API communications. You just can't expect user sto come to your site, visit your blog to keep up to date, and a feed for any blog instantly makes it portable and able to go to where your users are.
- **Twitter** - Twitter as a communication platform, much like a blog is a great way to establish an active presence for an API, providing updates about API endpoints, build relationships with developers and establish partnerships with other API providers. Also in line with a blog, it can be the communication tool that demonstrates your supporting your API, while an out of date Twitter stream can show that nobody is home to support an API--sending the signal developers should steer clear of the platform.
- **Email** - Sometimes your API building blocks are not complex tools or documentation. They can be as simple as an email address. Of course simply listing an email address as part of your API community is not where it ends. Don't list an email for developers, partners, or support if nobody answers it. Make sure and have a plan for any email addresses you use throughout your API community. Make them meaningful and route to the right people. Make sure your email accounts are responsive, otherwise people will immediately go to your forums with negative feedback. Consider an email account to provide support for your API community.
- **LinkedIn** - LinkedIn is a powerful business communications platform. While LinkedIn is not the preferred platform of many open API developers, it is the preferred platform of enterprise developers. As an API communications building block, an active presence on LinkedIn is recommended for API owners, it can add a healthy dimension to your communication strategy and reach older, more established developers that may not always be considered when deploying public APIs.

- **Facebook** - Like LinkedIn, Facebook carries a great deal of social weight when it comes to working with developers. Depending on your target developer audience, Facebook may or may not make sense as part of your communication strategy. Facebook is larger than just the individual social network accounts, and a Facebook Page can be a great way for API owners to attract and engage with developers who are building applications. Consider the Facebook effect when assembling API communication building blocks.
- **Google+** - While Google+ is not as popular as Twitter or Facebook, it does have as many active users as LinkedIn these days, and with considerable SEO benefits, it is recommended that you consider Google's social network as one of the API communication building blocks. Google+ has a tremendous amount of network effects beyond just the social network and Google SEO. Tools like Google Hangouts can be used as part of API open office hours, and events can be used to coordinate API focused gatherings.
- **Email Newsletter** - An email newsletter is a proven communication tool beyond APIs, and while many developers will not be open to receiving regular emails about an API, there are some developers who are still receptive to this format. As an API owner, there is also a positive effect from having to gather thoughts each week for an email newsletter that goes beyond just communicating with the developer community.
- **Instagram** - Using Instagram as a way to communicate with platform users, and let users know there is someone home.
- **Vimeo** - Using Vimeo as a communication channel for educating users about an API platform.
- **Youtube** - Using Youtube as a regular content and communication channel to help users learn about platform operations.
- **Chat** - Providing a live chant mechanism in the developer portal, allowing developers to connect directly with API providers during regular business hours, or specifically defined office hours.

Updates

Adding to the communication flow, standard updates further enhance overall awareness of platform operations throughout an ecosystem. A handful of common building blocks have been employed to provide consistent API platform updates.

- **Status Dashboard** - API owners are asking developers to invest in building applications on their platform. This is asking for a lot of trust, and the best way an API owner can build this trust with its developers is with a transparent roadmap. API roadmaps are usually a simple, bulleted list, derived from the APIs own internal roadmap, showing what the future holds for the platform. Transparency around an APIs roadmap is a tough balance, since you don't want to give away too much, alerting your competitors, but your developer ecosystem needs to know what's next. API owners need to find a balance that works for their company, and maintain an active roadmap outlining where the platform is headed.
- **Roadmap** - API owners are asking developers to invest in building applications on their platform. This is asking for a lot of trust, and the best way an API owner can build this trust with its developers is with a transparent roadmap. API roadmaps are usually a simple, bulleted list, derived from the APIs own internal roadmap, showing what the future holds for the platform. Transparency around an APIs roadmap is a tough balance, since you don't want to give away too much, alerting your competitors, but your developer ecosystem needs to know what's next. API owners need to find a balance that works for their company, and maintain an active roadmap outlining where the platform is headed.
- **Change Log** - Knowing the past is a big part of understanding where things are in store for the future. A change log should work in sync with the API roadmap building block, but provide much more detailed information about changes that have occurred with an API. Developers will not always pay attention to announced updates, but can use a change log as a guide for deciding what changes they need to make in their applications and how they will use an API. The change log will be another building block to keep developers updated, and reduce overall support resources needed.
- **Status RSS** - Providing an RSS feed for a status dashboard, allowing users to receive updates on any website, and via any RSS client.

Service Levels

APIs aren't just tech, they are the next generation of business development. Not all APIs have a clear business model, but for those that do, you find thesomese common elements between them, allowing them to successfully monetize their API driven digital resources.

- **Pricing** - Pricing doesn't always apply to APIs. It's very common to provide API service for free. However, whether or not you charge for an API, you should clarify this for developers. Provide a pricing page, outlining what a developer gets for free

and provide clear pricing for any other service levels, so developers will know what to expect as their usage grows. Even if the API is free, API owners should put thought into the future of the platform, set realistic expectations of how the platform will generate revenue to say in operation.

- **Service Tiers** - A well planned API will have multiple service tiers for developers to take advantage of. Before developers begin integrating their applications with an API, they need to have a clear understanding of what services are available to them. Successful API owners need to openly communicate all service tiers available, and provide simple and comprehensive descriptions of each. With no surprises on services available to them, developers can confidently build their applications on top of an API, understanding at which levels they will need to adjust their integration to take advantage of new levels of a platform.
- **Partner Program** - Adding a top level tier to an API program, establishing a transparent partner level for API developers to aspire to. Clearly sharing what it takes to become a partner, what the benefits of partnership are, and potentially list out all existing partners.
- **Reseller** - A formal program for managing resellers of products and services sold via APIs.
- **Volume Pricing** - Providing additional layers of pricing based upon volume purchases, allowing users to better understand their API usage and potentially buy in bulk based upon their past habits.
- **Certification Program** - Allowing developer to get certified, providing a verified layer of developers that can be used by the platform, and ecosystem partners.

Additional Resources

There will be different ways that API consumers learn about an API. Some of us rely on documentation, some on ready to go SDKs, while others will rely on videos, how-to guides, or possibly case studies. Many common resources are used to help users understand, as well as stay in engaged with an API, and its operations.

- **Case Studies** - APIs are all about partners and developers building new applications and finding innovative ways to integrate. When anyone builds some notable application on top of an API, develop a case study. Case studies don't need to be novels, make them short, concise and showcase what a partner or individual developers has done. Case studies will stimulate other developers imaginations, while also showing the API is a viable platform that others are building on top of.

- **How-to Guides** - Many developers can get up and running without any help at all. Other developers need a helping hand, showing them how to use the API and put code samples and SDKs to use. How-to guides can provide the essential resources for developers to get up and going with an API. Start with common integration scenarios and build how-to guides around them, then as new ways of integrations emerge create fresh how-to guides using these new ways of taking advantage of the API.
- **White Papers** - White papers demonstrate domain and industry expertise. APIs are about exposing valuable business resources and assets of a company. Producing white papers can actively demonstrate the expertise a company possess and how the API resources the company offers can solve problems and provide sounds solutions for an industry and business sector. Make white papers a regular part of the API content creation, and when ready, publish to the API area as well as syndicate across the web.
- **Webinars** - Not everyone likes to read how-to guides or case studies. Many developers prefer to have a visual walk through of how to integrate with an API or the case studies of how other developers have built on top of an API. When appropriate, make webinars and videos around your how-to guides and case studies. If video productions of case studies and how-to guides are standard operating procedure, the work can occur while you produce the core paper. Youtube and Slideshare are great platforms for distributing webinars and videos of API resources.
- **Events** - Publish a page with information about any events the an API program will be attending, or maybe links to video and slide decks from past conferences.
- **Slideshare** - Publish a list of slide decks available on a Slideshare account, providing developers with access to past talks the API program has done.
- **Codecademy** - Creating courses on Codecademy, then publishing them within an API developer area, providing lessons for developers on how an API operates.
- **Tour** - Providing a guided tour of how a platform and API operates, allowing new developers to walk through all aspects of operations in an interactive or video format.
- **Glossary** - A glossary of terms that applies to an industry, platform, and the business or technical aspects of operations, providing API consumers who may not be familiar with sector or platform specific terms, with a dictionary they can use during integration.
- **Videos** - Provides video resources, helping people understand the API platform, how it can be put to use, and any other supporting areas of API operations.

- **Internationalization** - Provide resources that help developers localize, and internationalize their applications to meet the needs of specific global markets.

Research & Development

When a company finds its footing with API operations, it can quickly go beyond just business development, it can open up an external research & development area for a company or institution. There are several building blocks employed to help turn APIs into the innovation engines that they can be.

- **Labs** - A labs environment for an API can be a center of innovation and inspiration for your API ecosystem. A labs environment usually showcase experimental and non-production projects built around your API. Labs can showcase experimental and research development by your internal development and business staff.
- **Ideas** - As an API owner you will have ideas flowing from all directions--internally, from partners and submissions from the API community. Establishing a revolving door for ideas is important, if they don't take hold internally and immediately get used, put them out to the community and showcase them in an idea forum. Encourage developers to submit their own, vote on, and comment or take ownership of ideas. Idea showcases stimulate developers, planting the seeds of innovation every API owner wants to see thrive in their ecosystems.
- **Opportunities** - After you've managed developer communities for a while you will find developers are very open to suggestion and pointing them in the right direction. Some API owners are creating sections of their API communities that showcase opportunities for developers to start projects and build applications. A dedicated opportunities page, bundled with an idea showcase and a transparent roadmap will help keep developer activity in line with company goals.

Legal Department

There are three tenants to API operations, the technology, the business, and the politics, which the legal side of operations and conducting business plays a huge role. The following building blocks are some of the most important aspects of API operations, and guiding many of the decisions that occur in the online, digital world we have created for ourselves.

- **Terms of Service** - Provides a legal framework for developers to operate within. They set the stage for the business development relationships that will occur within an API ecosystem. TOS should protect the API owners company, assets and brand,

but should also provide assurances for developers who are building businesses on top of an API. Make sure an APIs TOS pass inspection with the lawyers, but also strike a healthy balance within the ecosystem and foster innovation.

- **Privacy Policy** - Privacy policies protect the rights of partners, developers and platform users while also protecting the API owner from damaging activity via the API platform. Like API terms of use, privacy policies need to strike a balance and protect everyone involved, but also allow for innovation and commercial activity.
- **Branding** - Along with the other business assets made available via an API ecosystem, the API owners brand is also being put on the on the line. Branding Guidelines set the tone for how partners and developers can use the resources and assets made available via an API. The branding guidelines will provide a framework for attributing the API owner, how resources can be displayed and provide visual assets to support the company brand. As with terms of use and privacy policies, branding guidelines need protect the API owner but also provide developers with enough freedom to innovate.
- **Code License** - You should make sure all your code samples, libraries and SDKs have a default license that helps defines how your code can be used. A license for the code sets the expectations with API consumers regarding how they can use code generated by a company, and also protect API providers and their partners intellectual property. I recommend taking a look at Github's open source licensing page for a nice overview of the options, especially since this is one possible location you will be providing access to your code.
- **Data License** - A data license defines how data resources available via an API can be used. A data license sets the expectations with API consumers regarding how they can use data, and also protects API providers and their partners intellectual property. I recommend taking a look at Open Definition for an assortment of data and content licensing.
- **Service Level Agreement** - A service level agreements (SLA) provides a legal framework for describing what service(s) is being offered by an API provider, with details about level and quality of service, including warranties, disaster recovery, and steps for termination of agreement. SLAs may vary based upon multiple levels of access to API resources, and different API user groups.
- **Deprecation Policy** - An API deprecation policy sets expectations with API consumers about when and how API resources will be deprecated and shut down. These policies help build trust with API consumers, giving them an idea of how much

they can depend on an API resource, and what they can expect when it reaches the end of life.
- **Monetization Guidelines** - Providing legal guidelines for how developers can monetize around resources that are available via an API.
- **Compliance** - A page providing compliance related information, covering how a company, platform, and APIs complies with government and industry regulations.
- **Software License** - A licensing applied to the server or other software.
- **Trademarks** - Providing a reference of corporate trademarks that are part of API operations, and developers may have to consider.

Embeddable Tooling

APIs power a lot of things, but one of the potentially most ubiquitous aspects of any API is its embeddable tools. There are many building blocks in this area that have led the charge to API success. Think about the power of the Twitter retweet button, or the Facebook like -- API driven, embeddable building blocks were at play.

- **Widgets** - Widgets are highly functional, API driven JavaScript tools that provide portable applications that can be embedded on any website or application. API widgets provide tools any API user can deploy, and developers can reverse engineer, modify and extend to meet their needs. Widgets really establish an advanced API embeddable strategy and can deliver the value of an API across the Internet.
- **Badges** - Badges are common for displaying content and resources delivered via an API and allow these assets to be embedded on any website or application. API platforms like LinkedIn and Google have successfully employed API driven profile badges allowing any user to take advantage of the power of an API, and grow a healthy API embeddable strategy.
- **Widget Builder** - A widget builder provides a form that API consumers can use to configure a predefined widget that is integrated with an API, allowing them to customize and tailor the widget experience. A widget builder goes well beyond just a single widget, and gives embeddable code more of an application feel.
- **Buttons** - Buttons are shareable snippets of code that often share, syndicate or trigger a variety of events that benefit an API platform. Buttons play an important role in social media and social networks. Consider how Twitter's share button has made Twitter a global communication platform or how the Digg button transformed social

news. Embeddable buttons built on top of an API can significantly extend the reach of API resources.

- **JavaScript API** - A complete JavaScript API that developers can use to build and manage their own JavaScript integrations with an APIs resources.

Environment

Some areas where APIs are making an impact, cannot be safely developed against in a direct production environment. Many mission critical APIs have developed several building blocks, that help developers build, test, secure, and scale applications without making a negative impact on the experience.

- **Sandbox** - With the sensitive information available via many APIs, providing developers a sandbox environment to develop and test their code might be wise idea. Sandboxes environments will increase the overall cost of an API deployment, but can reduce headaches for developers and can significantly reduce support overhead. Consider the value of a sandbox when planning an API.
- **Production** - When planning an API, consider if all deployments need to have access to live data in real-time or there is the need to require developers to request for separate production access for their API applications. In line with the sandbox building block, a separate API production environment can make for a much healthier API ecosystem.
- **Simulator** - Providing an environment where developers can find existing profiles, templates or other collections of data, as well as sequence for simulating a particular experience via an API platform. While this is emerging as critical building block for Internet of Thing APIs, it is something other API providers have been doing to help onboard new users.
- **Templates** - Predefined templates of operation, when a new environment is setup, either sandbox, production, or simulator, it can be pre-populated with data, and other configuration, making it more useful to developers. These templates can be used throughout the API lifecycle from development, QA, all the way to simulation.

Developer Account

API consumers need visibility into their engagement with any platform. Augmenting the building blocks outlined as part of on-boarding, there are many common building blocks also emerging

to assist developers in managing their accounts, understand their usage, dial-in their billing, and stay engaged with an API platform.

- **Developer Dashboard** - Much like the landing page for the entire API platform, developers should have a single dashboard for getting at all their tools, metrics and information they need to successfully manage their usage. Developers should not have to ask, or look around for their account and access information--they should have a single place to obtain what they need.
- **Account Settings** - Along with password reset, access to their basic account detail and settings is standard operating procedure for any platform. Don't make developers look for their settings, give quick access to settings and allow for easy updates. If developers do not have access to change their settings themselves, they will be asking you for assistance requiring additional resources.
- **Reset Password** - The need to reset an account password access is pretty standard operations for any online platform. Provide the necessary tools for developers to gain access to their account if they lose their password.
- **Application Manager** - Many popular APIs are becoming application centric and provide developers with tools for managing multiple applications or development projects. API owners should consider how developers will be building applications on top of an API and consider that many will need multiple access keys for their separate applications or user groups.
- **Usage Logs & Analytics** - Rate limiting will be part of any API platform, without some sort of usage log and analytics showing developers where they stand, the rate limits will cause nothing but frustration. Clearly show developers where they are at with daily, weekly or monthly API usage and provide proper relief valves allowing them to scale their usage properly.
- **Billing History** - Obviously if an API is entirely free, billing history is not necessary, but if any tier of API requires paid access, provide clear and easy access to what a developer has been billed, allowing them to access and download their billing history. Provide tools for developers to update their billing and account information easily, as well as support for their billing questions.
- **Message Center** - Providing developers with a messaging system within their developer accounts, and communicate with API providers, and receive system updates.

- **Github Authentication** - Allow developers to create and login to their API developer account using their own Github account. It is easy to allow Github oAuth to be used, in place of making developers create yet another account.
- **Delete Account** - Developers have the ability to delete their account from within their account portal.
- **Service Tier Management** - The ability to manage your account service tier within a password protected portal, giving developers the ability to change their pricing, and service tier.

Reciprocity

APIs enable an unprecedented amount of interoperability, if implemented properly. More API providers are supporting reciprocity with other API providers, using services like IFTTT and Zapier, to facility integration, interoperability, and automation. Here are some of the common patterns I'm seeing emerge, that are responsible for this occurring.

- **Terms of Service** - The Terms of Service (TOS) is the central hub which makes the API economy work (or not work). TOS is where the protections for platform owners, developers and end-users exists. Restrictive TOS can suffocate the reciprocity of platform, while more sensible ones allow for the movement, and collaboration around resources that will make a platform thrive.
- **Data Portability** - Providing users with the ability to get data out of a system through a bulk download and via an API is essential to reciprocity existing. Along with other basic web literacy skills that every user should possess, every person should demand that any services they sign up for, should allow for data portability of all their resources.
- **Automation** - Providers like Zapier and IFTT are delivering API automation services for hundreds of popular APIs, allowing developers and end-users to further automate their operations across multiple platforms, allowing anyone to better manage their resources using very simple API driven workflows.
- **oAuth** - While not a perfect standard, oAuth is the best we have when it comes to providing an identity and access layer for API driven resource, one that allows for reciprocity to occur within a single API ecosystem, and between multiple ecosystems. oAuth gives the platform, developer and end-users a (potentially) equal role in who has access to API driven resources, governing how reciprocity is realized.

- **Integrations** - Integrations showcase other 3rd party platforms that an API provider is already connected to, providing ready to go platform integration solutions that any developer can take advantage of. This is right between reciprocity tools like Zapier, and not quite platform development kits (PDK).
- **Zapier** - Provides default Zapier integration, and usually is showcasing it within an API or developer area.

Security

Security is priority number one with APIs. Make sure and employ common building blocks like SSL and OAuth, as well as provide a simple overview that gets API consumers up to speed. Common security building blocks don't just help API consumers, they also make sure API providers have their ass covered, for the coming storm.

- **Security Overview** - An overview page outlining the security practices, and technology for a platform. Being upfront with developers about security practices of the platform, and what is expected of developers, leads to healthier, and stable platform operations.
- **SSL** - Providing SSL for API endpoints, and developer account, ensuring that end to end encryption is available for all aspects of API integration.
- **oAuth** - Using oAuth to secure an API, providing identity and access management for not just platform developers, but also allowing users to have some control over their own content and data that is accessed via an API.

Rate Limits

Every API platform will need to impose rate limits on API consumers, however it is how this is communicated, and shared that makes the difference between smooth API operations, and very rough integration experience. There are a handful of common ways to communicate around rate limits, when it comes to API operations.

- **Rate Limits Page** - Developers need to understand what the limitations of API usage are upfront. Rate limiting and throttling of API access has become commonplace to manage security, and resource load. This building block is about providing a single page explaining how this process works, setting expectations with API consumers.
- **Rate Limit Information In-line In Docs** - In addition to a rate limits page, explaining information about limitations on API usage, some API providers like Twitter are

including rate limits within API documentation. With this approach, each API endpoint has its related rate limits published alongside other details.

- **Account Rate Limit API** - As API usage grows the need to be able to programmatically manage your account is increasing. Some API platforms also provide APIs for the management of platform operations. It makes sense that API providers should also offer an API for getting rate limit details for their account.

Management API

As the number of APIs grows, so does the number of APIs any one developer is depending on, making the need for automation in API account management become a higher priority. An increasing number of API providers are providing APIs to their own platform user, and app management solutions. These are just a few of the common management APIs I am seeing emerge.

- **User Management** - Allow API API consumers to manage their own accounts via an API management API, enabling users to create, read, update, and delete information associated with their account--fields may vary, depending on what information each API requires for user accounts.
- **Account Management** - Allow API API consumers to manage their own accounts via an API management API, enabling users to create, read, update, and delete information associated with their account--fields may vary, depending on what information each API requires for user accounts.
- **Application Management** - Ideally each user can have multiple applications, consuming API resources at various rates. This allows for the most flexibility in API consumption, but may vary depending on what API management infrastructure employed. This API should allow for management of all applications, with secure control over application keys. Additionally, there should be analytics available, with a short, simple, but robust list of metrics.
- **Service Management** - Enable API consumers to retrieve information about plans available for a specific API, or stack of APIs. Allow for listing of plans, and the features available for each plan. Also recommend considering the ability to set default plan for an account, enabling smooth application management.

That concludes the most common building blocks I have aggregated from looking at thousands of API operations. I track on a number of other ones I see emerging, or employed in niche

scenarios, but this represents the common blueprint across most API providers you will come across today.

Do It Yourself (DIY) API Management

Tackling API management on your own is a realistic strategy. There are many companies and even individuals who manage one or many APIs by designing and deploying their own interfaces and cobbling together a management platform using common cloud services and open sources tools.

While a robust, API portal, complete with service composition, billing, analytics and other tools is optimal, a DIY approach is a cost effective approach that can get you to proof of concept and production without added costs of software and contracts.

It can be very difficult to anticipate what will happen in the deployment and operations of an API. While you can establish a base strategy for your API, most initiatives are real-time research and development around business data and resources, something a DIY approach is well suited for.

API service providers tend to focus on delivering tools for you to secure, manage developers, analytics and billing solutions. These are all very valuable tools for any API provider, so make sure and consider using one of the providers alongside a DIY approach. It is not necessarily a DIY vs. API management service provider question.

The Growing World of API Management Services Providers

Web API management as a service has been going on for some time now, and combined with the wealth of SOA experience among the providers, you can be pretty sure that the companies who have been in the business for a while, know what they are doing.

The first wave of API experience bame out of the enterprise, but quickly a new breed of web API focused companies emerge who redefined the way you manage APIs, with an emphasis on open and public.

In 2015, there are now over 40 API management providers ranging from the classic enterprise focus, to data to API solutions in the cloud. While there are differences between these providers, they all have a great deal of experience in planning and execution of API management for a wide variety of API resources.

I'll start by showcasing some of the service providers who offer a speciality focus, then move into other more general API management providing solutions for startups, up to the enterprise.

The API Portal

All good APIs start with an intuitive, easy to access portal. The quickest way to get up and going with a portal is using Github Pages, after that there a number of other cloud-based services dedicated to deploying an API developer portal for your operations.

- **3scale** - 3Scale provides plug-and-play as well as enterprise level API management services. 3Scale is similar to 3Scale connect is a starter platform with a freemium model for delivering your API. You can deploy at no cost, and pay-as-you-go based upon the volume of calls made on your API. This model is well suited to those who are not sure of their API business model or target audience — or are just looking to test the waters

    - **Website:** http://www.3scale.net
    - **Twitter:** https://twitter.com/3scale
    - **Blog:** http://www.3scale.net/blog/
    - **Github:** https://github.com/3scale

- **Gelato** - Gelato.io is a tool focused on creating API Documentation and Developer Portals. You can describe your API, import and sync with Swagger or API Blueprint, have your developers register and get updates when things change, get an automatic API Explorer.

    - **Website:** https://gelato.io/

- **Github Pages** - GitHub Pages are public web pages hosted and easily published through GitHub. The quickest way to get up and running is by using the Automatic Page Generator to create some starter HTML and CSS for us. You can then modify our GitHub Pages' content and style remotely via the web or locally on computer.

    - **Website:** https://pages.github.com/

- **Readme.io** - ReadMe is a developer hub for your startup or code. It's a completely customizable and collaborative place for documentation, support, key generation and more. Don't waste your precious time reimplementing a dev.yourstartup.com. If you have an API, code library, or affiliate program, documentation is hard. Users can't use your product without understanding it, and most documentation on the Internet is

lacking. The goal of ReadMe is to make consuming APIs completely painless.

    - **Website:** https://readme.io/
    - **Twitter:** https://twitter.com/readmeio

Documentation

Documentation is important, and there are some pretty slick services emerging that help you manage the documentation for your API.

- **Dexy** - Dexy helps your code to speak for itself. Show off your code with beautiful syntax highlighting. Write examples and Dexy will run them, inserting the output into any document you wish. Everything is based on live code, so updating is easy, syntax errors blow up on you, not your users, and typos are a thing of the past. With Dexy's smart caching, your code is only executed when it needs updating, saving you time while keeping your documents robust.

    - **Website:** http://dexy.it/
    - **Blog:** http://blog.dexy.it/
    - **Github:** https://github.com/dexy

- **Gelato** - Gelato.io is a tool focused on creating API Documentation and Developer Portals. You can describe your API, import and sync with Swagger or API Blueprint, have your developers register and get updates when things change, get an automatic API Explorer.

    - **Website:** https://gelato.io/

- **Readme.io** - ReadMe is a developer hub for your startup or code. It's a completely customizable and collaborative place for documentation, support, key generation and more. Don't waste your precious time reimplementing a dev.yourstartup.com. If you have an API, code library, or affiliate program, documentation is hard. Users can't use your product without understanding it, and most documentation on the Internet is lacking. The goal of ReadMe is to make consuming APIs completely painless.

    - **Website:** https://readme.io/
    - **Twitter:** https://twitter.com/readmeio

Communication

These may seem like obvious solutions, but they are important services for helping you manage communication with API consumers, that you may already be using.

- **Blogger** - Blogger is a blog-publishing service that allows multi-user blogs with time-stamped entries. It was developed by Pyra Labs, which was bought by Google in 2003. Generally, the blogs are hosted by Google at a subdomain of blogspot.com. Blogs can also be hosted in the registered custom domain of the blogger (like www.example.com). So blogspot.com domain publishings will be redirected to the custom domain. A user can have up to 100 blogs per account.

    - **Website:** http://www.blogger.com/
    - **Blog:** https://code.blogger.com/

- **Tumblr** - Tumblr is a re-envisioning of tumblelogging, a subset of blogging that uses quick, mixed-media posts. The service hopes to do for the tumblelog what services like LiveJournal and Blogger did for the blog. The difference is that its extreme simplicity will make luring users a far easier task than acquiring users for traditional weblogging. Anytime a user sees something interesting online, they can click a quick Share on Tumblr bookmarklet that then tumbles the snippet directly. The result is varied string of media ranging links and text to pictures and videos that takes very little time and effort to maintain.

    - **Website:** https://www.tumblr.com/
    - **Twitter:** https://twitter.com/tumblr
    - **Blog:** http://staff.tumblr.com/
    - **Github:** https://github.com/tumblr

- **Twitter** - Twitter is a global real-time communications platform with 400 million monthly visitors to twitter.com, more than 200 million monthly active users around the world.

    - **Website:** https://twitter.com/
    - **Twitter:** https://twitter.com/twitterapi/
    - **Blog:** https://dev.twitter.com/blog/
    - **Github:** https://github.com/twitter/

- **WordPress.com** - WordPress, which is commonly used to refer to all WordPress products, is the most popular and fastest growing publishing platform on the web. WordPress began as a blogging platform but soon evolved to include additional types of websites including news sites, corporate sites (for large brands and small businesses alike), e-commerce sites and everything in between.

    - **Website:** http://develop.wordpress.com/
    - **Twitter:** https://twitter.com/wordpressdotcom
    - **Blog:** http://wordpress.org/news/

Code Management

Github is king when it comes to code management in 2015. Managing all of your code from full blown repositories to snippets and libraries in Gists, should be done using Github.

- **GitHub** - GitHub is a social network for programmers. Git is a distributed software management program created by Linus Torvalds, originally for the Linux Kernel Development. GitHub is a hosted Git repository. Github allows you to take part in collaboration by forking projects, sending and pulling requests, and monitoring development.
    - **Website:** https://github.com
    - **Twitter:** https://twitter.com/github
    - **Blog:** http://github.com/blog
    - **Github:** https://github.com/github

- **Github Gists** - Gist is a simple way to share snippets and pastes with others. All gists are Git repositories, so they are automatically versioned, forkable and usable from Git.

    - **Website:** https://gist.github.com/

Self-Service Support

There are a handful of self-service support services I encourage you to look at when planning your API management.

- **Get Satisfaction** - Get Satisfaction is the community platform that helps companies create engaging customer experiences by fostering online conversations about their products and services at every stage of the lifecycle. Get Satisfaction powers 70,000

active customer communities hosting more than 35 million consumers each month. From its inception in 2007, Get Satisfaction has been focused on building an intuitive, easy to use community platform that is designed to bring resolution to consumers; is highly discoverable by search engines and is implemented quickly and easily.

    - **Website:** http://getsatisfaction.com
    - **Twitter:** https://twitter.com/getsatisfaction
    - **Blog:** http://blog.getsatisfaction.com/
    - **Github:** https://github.com/satisfaction

- **UserVoice** - UserVoice creates simple customer engagement tools that help companies understand and interact with their customers more positively and build customer relationships that last. UserVoice Feedback - a hosted tool for gathering and prioritizing product ideas directly from a company's customers.

    - **Website:** https://uservoice.com/
    - **Twitter:** https://twitter.com/uservoice
    - **Blog:** https://community.uservoice.com/blog/
    - **Github:** https://github.com/uservoice

Direct Support

When it comes to direct support with API consumers, Zendesk is definitely where it is at.

- **Zendesk** - Zendesk provides an integrated on-demand helpdesk - customer support portal solution based on the latest Web 2.0 technologies and design philosophies. The product has an elegant, minimalist design implemented in Ruby on Rails and provides seamless integration of the back-end helpdesk SaaS to a company's online customer-facing web presence, including hosted support email-ticket integration, online forums, RSS and widgets. This is unusual, because most SaaS helpdesk solutions focus exclusively on the backend helpdesk and treat the Web as an afterthought.

    - **Website:** http://www.zendesk.com/
    - **Twitter:** https://twitter.com/zendesk
    - **Blog:** http://www.zendesk.com/blog
    - **Github:** https://github.com/zendesk

Open Source API Management Solutions

In 2015, there are now almost 10 viable API management solutions. The growth of open source in the area of API management is a sign the space has matured. These are the open source projects I am tracking on currently.

- **Apify** - Apify is a small and powerful open source library that delivers new levels of developer productivity by simplifying the creation of RESTful architectures. It helps development teams deliver quality web services and applications in reduced amounts of time. Web services are a great way to extend your application, however, adding a web API to an existing web application can be a tedious and time-consuming task. Apify takes certain common patterns found in most web services and abstract them so that you can quickly write web APIs without having to write too much code.

    - **Website:** http://www.apifydoc.com/
    - **Twitter:** https://twitter.com/api_fy
    - **Github:** https://github.com/apify
- **APIMAN** - The apiman project brings an open source development methodology to API Management, coupling a rich API design & configuration layer with a blazingly fast runtime. Flexible, policy-based runtime governance for your APIs. Offer the same API through multiple plans, allowing different levels of service to different API consumers. Use the rich management layer (REST API and separate UI) to manage/configure not only APIs but also the applications that consume them.

    - **Website:** http://www.apiman.io
    - **Twitter:** https://twitter.com/apiman_io
    - **Blog:** http://www.apiman.io/blog/

- **Deployd** - An open source platform. Design, build, and scale APIs for web and mobile apps in minutes instead of days. 1) No boilerplate. With one command, your API is up and running. It's an empty canvas waiting for you to add Resources. No boilerplate code necessary. 2) Resources - Deployd APIs are built of plug-and-play resources, such as Collection, which can easily be added and defined through the deployd dashboard. 3) Dashboard - Add and manage your API's resources through an intuitive web-based dashboard. 4) 1-step deploy - When it's time to deploy, easily deploy it yourself anywhere that can host a Node.js app and MongoDB.

    - **Website:** http://deployd.com/
    - **Twitter:** https://twitter.com/deploydapp
    - **Github:** https://github.com/deployd

- **Kong** - Kong was created to secure, manage and extend Microservices & APIs. Kong is powered by the battle-tested tech of NGINX and Cassandra with a focus on scalability, high performance & reliability. Kong runs in production at Mashape handling billions of requests to over ten thousand APIs.

    - **Website:** https://getkong.org/

- **OpenRepose** - Repose is an open-source platform that you can use to build stacks of reusable software components. These components can be leveraged by service developers to perform common API processing tasks. By using Repose's components rather than creating their own, service developers can focus on the unique features of their services. Repose can be used inside a service to perform API operations. It can also run on one or more separate servers as a proxy to one or more services. At its core, Repose is a proxy that allows services to use Enterprise Integration Patterns (EIP).

    - **Website:** http://openrepose.org/
    - **Twitter:** https://twitter.com/openrepose
    - **Github:** https://github.com/rackerlabs/repose

- **Project Overlord** - Any computer system, whether it is centralized or distributed, needs some form of governance, i.e., the act of monitoring and managing the system. Such governance may be as simple as ensuring only authorized users have access to capabilities (e.g. services), or as complex as guaranteeing the system and its components maintain a level of availability or reliability in the presence of failures or increased system load. In order to have successful governance, some form of management, monitoring and administration is required for these processes.

    - **Website:** http://www.projectoverlord.io
    - **Twitter:** https://twitter.com/overlordapiman
    - **Blog:** http://jboss-overlord.blogspot.com/
    - **Github:** https://github.com/governance

- **Tyk** - An open source, lightweight, fast and scalable API Gateway. Set rate limiting, request throttling, and auto-renewing request quotas to manage how your users access your API. Tyk supports access tokens, HMAC request signing, basic authentication and OAuth 2.0 to integrate old and new services easily. Tyk can record and store detailed analytics which can be segmented by user, error, endpoint and client ID across multiple APIs and versions. Integrate your existing or new applications with Tyk using a simple REST API, Tyk even support hot-reloads so you can introduce new services without downtime.

    - **Website:** http://tyk.io/
    - **Blog:** http://tyk.io/blog/

- **Varnish Software** - Varnish Software is the company behind Varnish Cache, a widely trusted open source web accelerator, significantly enhancing web performance for businesses online. We power major sites across all industry types, or businesses. Varnish Plus is our commercial suite, offering products for scalability, customization, monitoring and expert support services.

    - **Website:** https://www.varnish-software.com/
    - **Twitter:** https://twitter.com/varnishsoftware
    - **Blog:** https://www.varnish-software.com/blog

- **WSO2** - WSO2 is the lean enterprise middleware company. It delivers the only complete open source enterprise SOA middleware stack purpose-built as an integrated platform to support today's heterogeneous enterprise environments internally and in the cloud. WSO2's service and support team is led by technical experts who have proven success in deploying enterprise SOAs and contribute to the technology standards that enable them.

    - **Website:** http://wso2.com
    - **Twitter:** http://twitter.com/wso2
    - **Blog:** http://wso2.com/blogs/thesource/
    - **Github:** https://github.com/wso2

Primary Group of Providers

All the rest of the solutions I'm going to throw into a single bucket. As I get to know these solutions, or the companies themselves help me understand the features they offer, I will put into other buckets. Some of these areas, I am already carving off into other research projects such as an SDK project--for now, here are the other companies I'm keeping an eye on.

- **Akana** - Akana helps businesses accelerate digital transformation by securely extending their reach across multiple channels u2013 mobile, cloud and Internet of Things. Akana enables enterprises to share data as APIs, connect and integrate applications, drive partner adoption, monetize their assets, and provide intelligent insights into their business and operations.

    - **Website:** http://www.akana.com/
    - **Twitter:** https://twitter.com/akanainc
    - **Blog:** https://blog.akana.com/

- **Amazon Web Services** - Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create an API that acts as a front door for applications to access data, business logic, or functionality from your back-end services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any Web application.

    - **Website:** http://aws.amazon.com
    - **Twitter:** https://twitter.com/awscloud
    - **Blog:** http://aws.amazon.com/blogs/aws/
    - **Github:** https://github.com/amazonwebservices

- **ApiAxle** - ApiAxle is a proxy that sits on your network, in front of your API(s) and manages things that you shouldn't have to, like rate limiting, authentication and caching. It's fast, open and easy to configure. ApiAxle is different to the cloud based services such as Mashery in that it's intended to be installed within your LAN and be managed by you. This means you own your users, you own your data and you can more easily manage costs. ApiAxle is open-source. This means you can modify it as much as you like and contribute changes back. Others will do the same and

gradually the system will become all the better for it.

    - **Website:** http://apiaxle.com/
    - **Twitter:** https://twitter.com/apiaxle
    - **Blog:** http://blog.apiaxle.com/
    - **Github:** https://github.com/apiaxle

- **Apigee** - Apigee delivers an intelligent API platform to accelerate the pace of digital business. We help companies - from disruptive start-ups to the Fortune 100 u2013 use their enterprise data and services to create connected digital experiences for customers, partners and employees. This is digital business.

    - **Website:** https://apigee.com
    - **Twitter:** https://twitter.com/apigee
    - **Blog:** https://blog.apigee.com/front
    - **Github:** https://github.com/apigee

- **Apitite** - Apitite is a web application programming interface (API) creation and management service on the cloud. Apitite enables businesses to securely expose their existing data to their users, employees, and partners automatically. Through our web interface, clients can create web APIs by providing connection details to their datastore, specifying which data to expose, and listing who should be granted access to the data. In addition, clients can access usage analytics to observe how their users are consuming their data. Apitite takes care of the web hosting, security, and user access of each API.

    - **Website:** https://www.apitite.net/
    - **Twitter:** https://twitter.com/apititeapi
    - **Blog:** http://blog.apitite.net/

- **Axway** - Axway (Euronext: AXW.PA), acquired API Gateway vendor, Vordel, to complement its existing MFT and B2B Gateway products. Axway's suite of products enables enterprise to govern the flow of data within and across the edge of the enterprise, unlocking the tremendous value this can bring to business interactions. Axway API Management offers the enterprise-grade API management architecture with the security to protect sensitive data, control access and support integration to a wide range of on-premise and cloud-based applications.

    - **Website:** http://www.axway.com/vordel-products/
    - **Twitter:** https://twitter.com/axway
    - **Blog:** http://blogs.axway.com/

- **CA API Management** - CA API Management provides the tools any organization needs to integrate and expose legacy systems and applications as APIs. Onboard, enable and manage developers to create apps. Secure enterprise data to meet the toughest compliance and regulatory standards and choose which apps, developers and partners can access your APIs.

    - **Website:** http://www.ca.com/us/products/api-management.aspx
    - **Twitter:** https://twitter.com/cainc

- **Dell Boomi** - Dell Boomi, a business unit of Dell, is the first and only integration solution built in the cloud, to fully exploit the value of the cloud. Organizations of all sizes from small businesses to the largest global enterprises trust Dell Boomi to quickly connect any combination of cloud and on-premises applications. Leading SaaS players and enterprise customers such as salesforce.com, NetSuite, Taleo, oneworld, AAA, and NASDAQ rely on Dell Boomi to accelerate time to market, increase sales, and eliminate the challenges associated with integration.

    - **Website:** http://www.boomi.com/
    - **Blog:** http://blogs.boomi.com/bod/

- **Exicon** - Exicon is the AppCycle management company that simplifies the creation, deployment and management of mobile applications for some of the world's biggest companies allowing them to accelerate their success in mobile. Through its proprietary AppCycle Management platform, Exicon provides the services and tools to support customers in defining their application needs, finding the best developers, storing all components of the applications, reaching the end users for your application and measuring its performance.

    - **Website:** https://twitter.com/
    - **Twitter:** https://twitter.com/exicon
    - **Blog:** http://www.exiconglobal.com/blog/

- **Fiorano** - Fiorano powers real time, digital enterprises with a bimodal integration and API Management strategy that leverages the best of systematic (centralized,

high-control) and adaptive (federated, high-speed) approaches to deliver solutions across cloud, on-premise and hybrid environments. We provide microservice-centric platform infrastructure with sophisticated integration capabilities, Connecting Anything to Everything.

    - **Website:** http://www.fiorano.com/
    - **Twitter:** https://twitter.com/fioranoglobal
    - **Blog:** http://www.fiorano.com/blog/

- **Gluu** - Gluu provides open source authentication and API access management stack, called the Gluu Server, helps companies secure Web and mobile applications. The Gluu Server leverages standards such as OAuth2, OpenID Connect, UMA, SAML 2.0, and SCIM to enable federated single sign-on (SSO) and trust elevation. The Gluu Server is used by universities, government agencies, and companies to secure employee facing and consumer network services. Deployed on one or more dedicated servers and the IaaS platform of your choice, the Gluu Server improves the quality and drives down the cost of an increasingly complex and mission critical IT service: authentication and authorization (AA).

    - **Website:** http://www.gluu.org/
    - **Twitter:** https://twitter.com/gluufederation
    - **Blog:** http://www.gluu.org/blog/

- **IBM** - IBM API Management solution provides a complete set of web API capabilities to help you extend services from the back-office to new front-office engagements. It offers flexible deployment options, including capabilities for creating, proxying, assembling, securing, and scaling web APIs. IBM API Management provides detailed analytics and operational metrics to the business owner, as well as a customized developer portal for socializing the APIs and managing applications that can be used by developers.

    - **Website:** https://apim.ibmcloud.com/
    - **Twitter:** https://twitter.com/ibmapimgt
    - **Blog:** https://developer.ibm.com/api/blog/

- **JustAPIs** - JustAPIs is designed to help developers overcome existing limitations when it comes to building APIs. The JustAPIs solution is a high-performance, compiled executable that can run on any Linux, Windows, or Mac OSX based

hardware, from a single developer's laptop to large-scale, clustered production environments. With zero dependencies, JustAPIs can be installed instantly and includes sample APIs with familiar JavaScript-based business logic, so developers can be up and running with their own API server in minutes.

    - **Website:** http://justapis.com/
    - **Twitter:** https://twitter.com/justapis

- **Mashape** - Mashape provides an API editor, as part of their API management and discovery platform, allowing API providers to add, edit, and manage the details of an API design, while also managing the rest of API operations--from design to discovery and integration. Mashape editor is just a piece of the overall Mashape suite of API lifecycle management tooling.

    - **Website:** http://mashape.com
    - **Twitter:** https://twitter.com/mashape
    - **Blog:** http://blog.mashape.com/
    - **Github:** https://github.com/mashape

- **Mashery** - You want to make it fast and easy for developers to start building with your API. Mashery I/O Docs let you achieve that with a clean, powerful interface for executing live API calls right from your API documentation. With Mashery I/O Docs, your developers spend less time toggling, cutting, and pasting and more time coding great apps.

    - **Website:** http://mashery.com
    - **Twitter:** https://twitter.com/mashery
    - **Blog:** http://www.mashery.com/blog
    - **Github:** https://github.com/mashery

- **Microsoft Azure** - Simple and frictionless access for developers is key to the success of any API program. Minimizing the time it takes for a new developer to complete a transaction with your API is crucial. API Management generates great documentation and provides an interactive console that rapidly increases developer success.

    - **Website:** http://azure.microsoft.com/en-us/services/api-management/
    - **Twitter:** https://twitter.com/azure

    - **Blog:** http://azure.microsoft.com/blog/

- **Monarch API Manager** - An open source solution for quickly deploying, managing, and analyzing your APIs. Manage partners, APIs, and internal access from an administrative GUI. Control security policies and apply authorization at a granular level around your digital assets. Capture and view real-time usage statistics across partners and APIs. Create custom data via our Analytics API. Use this data to derive business insights and drive monetization. Allow users to share their data with applications using the OAuth 2.0 protocol. Protect your customers' sign in credentials. Empower them to revoke applications at any time.

    - **Website:** http://www.monarchapis.com/
    - **Github:** https://github.com/monarchapis

- **MuleSoft** - Mulesoft provides a cloud, and open source version of their API design editor, enabling API designers to craft APIs using the RAML API definition format, then publish to notebook, as well as manage through other aspects of the API lifecycle with other Mulesoft systems.

    - **Website:** http://www.mulesoft.com/
    - **Twitter:** https://twitter.com/mulesoft
    - **Blog:** http://blogs.mulesoft.org/
    - **Github:** https://github.com/mulesoft

- **Nevatech Sentinet** - Nevatech Sentinet is a flexible, lightweight and scalable API management platform that promotes integration through the use of SOA standards. It is designed to connect, mediate, and manage interactions between services across the enterprise or in the cloud. If your organization uses internally or externally-facing web services and APIs, cloud-based infrastructure, or service-oriented architecture for your applications, then you will find Sentinet to be a powerful tool, which can be deployed rapidly, and immediately start delivering tangible results.

    - **Website:** http://www.nevatech.com/
    - **Twitter:** https://twitter.com/nevatechinc
    - **Blog:** http://www.nevatech.com/blog

- **Restlet** - The Restlet API Platform enables developers and non-developers to design, create, run and manage the APIs that provide access to any data or

application. Restlet Framework is the most widely used open source solution for Java developers who want to create and use APIs. The first Platform-as-a-Service dedicated to web APIs, APISpark enables any organization to become an API provider in minutes via an intuitive browser interface.

    - **Website:** https://restlet.com/
    - **Twitter:** https://twitter.com/apispark
    - **Blog:** http://blog.restlet.com/
    - **Github:** https://github.com/apispark

- **Sensedia** - Sensedia solutions help companies accelerate the integration between their internal applications and to carry vital data value chain partners to deal with security and control. With deep expertise and sensational tools for SOA (Service-oriented Architecture) and APIs (Application Programming Interfaces), many companies reduced from months to days the time needed to integrate with a new business partner or to connect applications that support business processes.

    - **Website:** http://sensedia.com/br/
    - **Twitter:** https://twitter.com/sensedia_br

- **SlashDB** - SlashDB connects your internal databases and constructs a REST/HTTP web service, easily making database content accessible by URLs for getting, updating, inserting and deleting in a secure way. SlashDB provides connectors for Microsoft SQL Server, Oracle, MySQL, PostGreSQL, IBM DB2 and Sybase--covering the top 5 databases you will find in the enterprise or small to medium businesses. SlashDB automatically turns databases into online resource so their content becomes accessible to authorized web, mobile and enterprise applications for reading and writing under standard data formats. Technically speaking, it makes REST APIs out of relational databases.

    - **Website:** http://www.slashdb.com/
    - **Twitter:** http://twitter.com/slash_db

- **Socrata** - Socrata is the leading developer and provider of Open Data Services, a category of cloud-based Web 2.0 solutions that enable federal, state, and local governments to dramatically improve the reach, usability and social utility of their public information assets. The Socrata Social Data Platform is a turnkey information

delivery platform that reduces lifecycle management costs for government customers while boosting their ability to disseminate relevant information and data-driven services to a wide range of audiences including citizens, civic application developers, researchers, journalists and internal stakeholders.

    - **Website:** http://socrata.com
    - **Twitter:** https://twitter.com/socrata
    - **Blog:** http://www.socrata.com/tech-blog/
    - **Github:** https://github.com/socrata

- **StrikeIron** - StrikeIron is the leader in Data-as-a-Service (DaaS), delivering data quality and communications solutions via our cloud platform IronCloud. We provide address verification, email verification, phone validation, phone append, SMS text messaging, and sales tax solutions to customers in a variety of markets. Our solutions are delivered as Web services that can be easily integrated into any application or system. Additionally, our solutions are pre-integrated into leading platforms like: Salesforce.com, Magento, Informatica, Oracle CRM On-Demand, and more.

    - **Website:** http://www.strikeiron.com/
    - **Twitter:** https://twitter.com/strikeiron
    - **Blog:** http://blog.strikeiron.com

- **StrongLoop** - StrongLoop develops StrongLoop Suite, a leading Mobile API Tier along with being the primary code contributor to Node.js. StrongLoop Suite includes an open source private mBaaS, an operations console and a supported package of Node.js, containing advanced debugging, clustering and support for private npm registries. StrongLoop was founded by Node core-contributors, Enterprise mobile architects and veterans of open source and Could companies.

    - **Website:** http://strongloop.com/
    - **Twitter:** https://twitter.com/strongloop
    - **Blog:** http://strongloop.com/strongblog/
    - **Github:** https://github.com/strongloop

- **SwiftIQ** - SwiftIQ provides web-service application programming interface (API) infrastructure to facilitate data accessibility and predictive analytics through the Swift Access and Swift Predictions products. Swift Access is an award-winning backend

platform to unify and secure disconnected data then deliver and analyze it on-demand to power real-time digital actions. Swift Predictions allows users to apply adaptive, machine learning algorithms to discover insights fast and make applications smarter.

    - **Website:** http://www.swiftiq.com/
    - **Twitter:** https://twitter.com/swiftiq
    - **Blog:** http://www.swiftiq.com/blog
    - **Github:** https://github.com/swiftiq

- **TIBCO Software** - TIBCO API Exchange Gateway governs third-party access to enterprise SOA systems by federating heterogeneous services and providing a single point of control. It lets you extend your enterprise applications and services onto cloud, partner ecosystem, and mobile platforms by managing and enforcing policies such as security, throttling, transformation, routing, and monitoring.

    - **Website:** http://www.tibco.com/
    - **Twitter:** https://twitter.com/tibco
    - **Blog:** http://www.tibco.com/blog/

- **WaveMaker, Inc.** - WaveMaker, Inc. provides WaveMaker Enterprise u2013 An aPaaS software for rapid application delivery of enterprise custom apps. WaveMaker Enterprise provides benefits such as visual rapid application development, out of the box support for security, web services integration and data modelling, high performance cloud platform based on Docker containers, a self service management console that provides simplified administration and many more such features.

    - **Website:** http://www.wavemaker.com/
    - **Twitter:** https://twitter.com/wavemaker
    - **Blog:** http://www.wavemaker.com/blog/
    - **Github:** https://github.com/wavemaker

API Management for the Internet of Things

As the need for APIs grows, and the concept of API management matures, it is natural for us to apply what we learn to the next wave of Internet growth--the space everyone is calling the Internet of Things. Recently I am seeing API management solutions that are dedicated to the IoT realm emerge, something I will keep an eye on to see how it grows.

- **JustAPIs** - JustAPIs is designed to help developers overcome existing limitations when it comes to building APIs. The JustAPIs solution is a high-performance, compiled executable that can run on any Linux, Windows, or Mac OSX based hardware, from a single developer's laptop to large-scale, clustered production environments. With zero dependencies, JustAPIs can be installed instantly and includes sample APIs with familiar JavaScript-based business logic, so developers can be up and running with their own API server in minutes.

    - **Website:** http://justapis.com/
    - **Twitter:** https://twitter.com/justapis

- **MachineShop** - MachineShop transforms the way enterprises and developers build, integrate and manage solutions to unlock more business value and intelligence. With MachineShop's API-centric middleware, every Internet-enabled thing, system, device, network, application, event and business process can be expressed as a service u2013 without the limitations of a proprietary Platform.

    - **Website:** http://www.machineshop.io/
    - **Twitter:** https://twitter.com/machineshopio
    - **Blog:** http://www.machineshop.io/blog

These are the companies I feel offer core API management solutions, and while some of them also assist in other areas like design, and deployment, these have to also lean heavily on providing management, or I don't include here.

There is a balance being struck between API service providers, and open tooling, when it comes to winning over the hearts and minds of API architects and designers. The smart companies have open source projects, as well as commercial services and tooling.

Beyond the Services There is Open Tooling

Open source tooling for API management was something that was pretty deficient in my opinion, up until the last couple years. However there are many open source solutions I consider pretty critical to making API management work.

This is just a small sampling of the tools I'm tracking on when it comes to managing APIs, now that I have a pretty robust blueprint of API management operations, from studying leading API

providers I will spend more time looking for alternate tools that can assist in delivering on these blueprints.

Communication Tooling

I have seen many good APIs start as a simple WordPress site, providing a simple portal to hang API document, code libraries, and provide communication and support. Never underestimate the power of a good WP installation to kick things off.

- **Wordpress** (http://wordpress.org/)- WordPress is an open source blog tool and publishing platform powered by PHP and MySQL. Its often customized into a Content Management System (CMS). It has many features including a plug-in architecture and a template system. WordPress is used by over 14% of Alexa Internet's top 1 million websites and is widely regarded as the most popular CMS on the internet according.

Tooling for Delivering API Documentation

There are a growing number of open source solutions for managing API documentation. This latest wave was set into motion by Swagger, with their UI interactive documentation, but has grown even larger than just the Swagger ecosystem, with other static, and dynamic API documentation solutions.

- **Carte** (https://github.com/devo-ps/carte)- Carte is a simple Jekyll based documentation website for APIs. It is designed as a boilerplate to build your own documentation and is heavily inspired from Swagger and I/O docs. Fork it, add specifications for your APIs calls and customize the theme. Go ahead, see if we care.
- **Flask API** (http://www.flaskapi.org/)- Flask API is an implementation of the same web browsable APIs that Django REST framework provides. It gives you properly content negotiated responses and smart request parsing.
- **RAML API Console** (https://github.com/mulesoft/api-console)- API Console is a graphical user interface for a RAML-defined API that visually exposes the API's structure and important patterns and serves as interactive API documentation. It is provided under the open-source CPAL license.
- **RAML to HTML** (https://github.com/kevinrenskers/raml2html)- A simple RAML to HTML documentation generator written for Node.js. RAML to HTML gets a RAML file and outputs the API documentation on a single HTML page.

- **Slate** (https://github.com/tripit/slate)- Slate helps you create beautiful API documentation. Think of it as an intelligent, responsive documentation template for your API. With Slate, the description of your API is on the left side of your documentation, and all the code examples are on the right side. Inspired by Stripes and Paypals API docs. Slate is responsive, so it looks great on tablets, phones, and even print.
- **Swagger UI** (http://swagger.wordnik.com/)- Swagger UI is part of Swagger project. The Swagger project allows you to produce, visualize and consume your OWN RESTful services. No proxy or 3rd party services required. Do it your own way. Swagger UI is a dependency-free collection of HTML, Javascript, and CSS assets that dynamically generate beautiful documentation and sandbox from a Swagger-compliant API. Because Swagger UI has no dependencies, you can host it in any server environment, or on your local machine.

An Open Source Option for Status Updates

There are a number of solutions out there for helping you publish the status of your API, but this was the first time I found an open source offering.

- **Stashboard** (http://stashboard.org/)- Stashboard is a status dashboard for APIs and software services. Its similar to the Amazon AWS Status Page or the Google Apps Status Page. Stashboard was originally written by Twilio to provide status information on its Voice and SMS APIs. Stashboard is designed to provide a generic status dashboard for any hosted service or API. The code can be downloaded, customized, and run anywhere.

Open Source Analytics for APIs

These are a few of the open source solutions I have found that have been used as part of API operations, providing valuable analytics about API usage.

- **Graphite** (http://graphite.wikidot.com/)- Graphite is a highly scalable real-time graphing system. As a user, you write an application that collects numeric time-series data that you are interested in graphing, and send it to Graphites processing backend, carbon, which stores the data in Graphites specialized database. The data can then be visualized through graphites web interfaces.
- **Stats D** (https://github.com/etsy/statsd)- StatsD is a dead simple NodeJS daemon that listens for messages over a UDP port. When a message is received, it parses

the messages, and extracts metrics data. StatsD lets you define what you want to track, throw it at the StatsD over a UDP port, once processed you can schedule it to dump to Graphite for further processing and visualization.

Finding Right Mix of Service Provider & Tooling For Your API Management

With API management, it often comes down to whether or not you have the skills and resources to tackle managing your entire operations without the assistance of an API management service provider--make sure you take advantage of some of the free / freemium services out there.

If you are part of a larger enterprise entity, you will most likely have options available to you through your existing IT Operations. But there is still a lot you can learn from the approaches by public API providers, and how they handle their API management, so don't close yourself off just because you are in the enterprise.

For most companies, when you are just learning about the API space, budgets will be tight, and you will have to cobble together what you need to get to a proof of concept. While just learning, experimenting and iterating, with an API, it can be tough to enter into contractual obligations for API management services. Be mindful mindful of costs while learning, the API management service providers can bring a wealth of experience to the table, but should understand your position, and not lock you into old school ways of selling software -- their pricing and way of doing business should reflect the API space.

No matter which path you choose, the DIY, or service provider, or a hybrid of both, it is highly recommended you maintain full control over API management across all parts of a business. API management is not just an IT or developer initiative. APIs are something that business and marketing departments need to be deeply involved with from design to management operations. It isn't something you should outsource entirely--just find 3rd party services to augment your own internal efforts.

You will notice there are some common services not listed above, like Google Groups. I do not recommend using Google Groups anymore because they do not have an API, for you to get your data out--I've seen projects run into serious trouble when trying to manage their ecosystem with the platform. You also notice that Stack Exchange is not listed here as well, because I do not recommend it be used as a core part of API management anymore, due to challenges other API providers have faced there--for now I leave it in the area of API evangelism.

With this research I hope I turned you on to many of the common building blocks used by leading API providers in the space, giving you a checklist to consider when planning your API

management strategy. Then I looked to introduce you to as many of the valuable services and tooling I know of in the sector, allowing you to make the decision for yourself around what will work with your company's approach.

My API management research is by far the most evolved of all of my research, resulting in these over 50 pages. However as the work evolves I keep carving off chunks, resulting in my API definitions, design, and deployment research. You will notice I will also be carving off SDK, status updates, and other areas into their own research projects as well. Ultimately all of these areas, when laid on the table, provide us with a venn diagram of the API economy, that we can use to position our own strategies--rooted in the best practices already in play across the industry.

API Management News from the Last Couple of Months

If you aren't familiar with what I do as the API Evangelist. I have spent the last five years monitoring the API space, curating links from across the web, and putting them into different buckets. This curation then goes into my research, analysis, and ultimate this white paper you are reading.

I've shared just the last 50 links, spanning the 2nd quarter of 2015. A complete list can be found on Github at management.apievangelist.com. While most of the links I provide are from 3rd party locations, some of them provide links to my own analysis in the area.

- **Many Leading API Management Vendors are Not So Secure After All** (08-14-2015 on www.axway.com) - https://www.axway.com/en/blog/2015/08/many-leading-api-management-vendors-are-not-so-secure-after-all-0
- **Make sure your Ads API client library runtime environment is up-to-date!** (08-12-2015 on googleadsdeveloper.blogspot.com) -http://googleadsdeveloper.blogspot.com/2015/08/make-sure-your-ads-api-client-library.html
- **Our Brand New Docs** (08-11-2015 on www.pushwoosh.com) - https://www.pushwoosh.com/our-brand-new-docs/
- **Amazon Release API Gateway, a Managed Service to Build and Run APIs** (07-12-2015 on www.infoq.com) - http://www.infoq.com/news/2015/07/aws-api-gateway
- **A first glimpse of the new Amazon API Gateway** (07-11-2015 on api-university.com) - http://api-university.com/blog/api-architecture/a-first-glimpse-of-the-new-amazon-api-gateway/
- **A first glimpse of the new Amazon API Gateway** (07-11-2015 on api-university.com) - http://api-university.com/blog/a-first-glimpse-of-the-new-amazon-api-gateway/
- **Amazon Web Services API Gateway: Why it could be a big deal** (07-10-2015 on www.zdnet.com) - http://www.zdnet.com/article/amazon-web-services-api-gateway-why-it-could-be-a-big-deal/
- **The New AWS API (Gateway): Anyone Who Does Not Do This, Will Be Fired. Thank You. Have A Nice Day! - Jeff Bezos ·** (07-09-2015 on apievangelist.com) -

http://apievangelist.com/2015/07/09/the-new-aws-api-gateway-anyone-who-does-not-do-this-will-be-fired-thank-you-have-a-nice-day--jeff-bezos

- **Amazon API Gateway – Build and Run Scalable Application Backends** (07-09-2015 on aws.amazon.com) - https://aws.amazon.com/blogs/aws/amazon-api-gateway-build-and-run-scalable-application-backends/
- **WSO2 to Present Webinar Series on API Design and Management Best Practices for Achieving a Connected Business** (07-09-2015 on wso2.com) -http://wso2.com/about/news/wso2-to-present-webinar-series-on-api-design-and-management-best-practices-for-achieving-a-connected-business/
- **Amazon API Gateway – Build and Run Scalable Application Backends** (07-09-2015 on aws.amazon.com) - https://aws.amazon.com/blogs/aws/amazon-api-gateway-build-and-run-scalable-application-backends/
- **AWS Launches API Gateway as a Cloud Service** (07-09-2015 on feedproxy.google.com) -http://feedproxy.google.com/~r/programmableweb/~3/7fyxydcfk0w/09
- **Managing apps and users with fine-grained access controls** (07-09-2015 on blog.heroku.com) -https://blog.heroku.com/archives/2015/7/9/managing_apps_and_users_with_fine_grained_access_controls
- **More Improvements to the Developer Docs Portal** (07-09-2015 on feedproxy.google.com) - http://feedproxy.google.com/~r/sforceblog/~3/tpf9fwlpfse/more-improvements-to-the-developer-docs-portal.html
- **Adding a user with Context.IO!** (07-08-2015 on blog.context.io) - http://blog.context.io/2015/07/adding-a-user-with-context-io/
- **BrandPost: About MaaS and PaaS of the future** (07-08-2015 on www.computerworld.com) - http://www.computerworld.com/article/2945396/enterprise-applications/about-maas-and-paas-of-the-future.html
- **New And Improved Developer Documentation** (07-08-2015 on blog.pokitdok.com) - https://blog.pokitdok.com/new-and-improved-developer-documentation/
- **The New AWS API (Gateway): Anyone Who Does Not Do This, Will Be Fired. Thank You. Have A Nice Day! - Jeff Bezos** (07-08-2015 on apievangelist.com) -

http://apievangelist.com/2015/07/09/the-new-aws-api-gateway-anyone-who-does-not-do-this-will-be-fired-thank-you-have-a-nice-day--jeff-bezos/

- **Manage Fuse APIs with apiman** (07-07-2015 on www.apiman.io) - http://www.apiman.io/blog/api/management/jboss/fuse/2015/07/07/fuse-apis.html
- **On ProgrammableWeb: Manage APIs with APISpark** (07-03-2015 on restlet.com) - http://restlet.com/blog/2015/07/03/on-programmableweb-manage-apis-with-apispark/
- **Use AlchemyData News Like a Pro: Tip #3** (07-02-2015 on blog.alchemyapi.com) - http://blog.alchemyapi.com/use-alchemydata-news-like-a-pro-tip-3
- **Yelp Announces API Console and Will Soon End V1 Endpoints** (07-02-2015 on www.programmableweb.com) -http://www.programmableweb.com/news/yelp-announces-api-console-and-will-soon-end-v1-endpoints/2015/07/02
- **How To Manage APIs with Restlet's APISpark** (07-01-2015 on www.programmableweb.com) - http://www.programmableweb.com/news/how-to-manage-apis-restlets-apispark/sponsored-content/2015/07/01
- **SLAs are just a comfort blanket. In the OnApp Federation, metadata rules!** (07-01-2015 on onapp.com) - http://onapp.com/blog/article/slas-comfort-blanket-metadata-rules-in-the-onapp-federation
- **What's New for Developers with QuickBooks Online v87?** (07-01-2015 on developer.intuit.com) - https://developer.intuit.com/blog/2015/07/01/whats-new-for-developers-with-quickbooks-online-v87
- **Sign-up For the Amazon Developer Weekly** (06-30-2015 on developer.amazon.com) -https://developer.amazon.com/appsandservices/community/post/tx1lfuawnpk9j6o/sign-up-for-the-amazon-developer-weekly
- **THBS Bolsters Its Flagship API Management Solution API-o-Blocks with Advanced Capabilities** (06-30-2015 on www.prlog.org) -http://www.prlog.org/12471152-thbs-bolsters-its-flagship-api-management-solution-api-blocks-with-advanced-capabilities.html
- **Data Request - Data.gov** (06-27-2015 on www.data.gov) - https://www.data.gov/data-request/request-id/6912/

- **Its all about managing those APIs better says WaveMaker** (06-26-2015 on cmsreport.com) - http://cmsreport.com/articles/it-s-all-about-managing-those-apis-better-says-wavemaker-13631
- **API v2 developer site (preview!)** (06-25-2015 on blogs.dropbox.com) - https://blogs.dropbox.com/developers/2015/06/api-v2-developer-site-preview/
- **API Management and Microservices** (06-25-2015 on www.windley.com) -http://www.windley.com//archives/2015/06/api_management_and_microservices.shtml#st_refdomain=&st_refquery=
- **Introducing the Alexa Skills Kit, Enabling Developers to Create Entirely New Voice Driven Capabilities** (06-25-2015 on developer.amazon.com) -https://developer.amazon.com/appsandservices/community/post/tx205n9u1ud338h/introducing-the-alexa-skills-kit-enabling-developers-to-create-entirely-new-voic
- **Announcing New Docs for Keen IO** (06-24-2015 on keenio.tumblr.com) - http://keenio.tumblr.com/post/122347324946/announcing-new-docs-for-keen-io#_=_
- **Focusing on Spot Instances – Let's Talk About Best Practices** (06-24-2015 on aws.amazon.com) - https://aws.amazon.com/blogs/aws/focusing-on-spot-instances-lets-talk-about-best-practices/
- **Someone Should Do A Site Like CodeProject, But Just For APIs ·** (06-23-2015 on apievangelist.com) - http://apievangelist.com/2015/06/23/someone-should-do-a-site-like-codeproject-but-just-for-apis/
- **How to Receive Notifications When Your AWS Accounts Root Access Keys Are Used** (06-23-2015 on blogs.aws.amazon.com) -https://blogs.aws.amazon.com/security/post/tx3pspqsn8374d/how-to-receive-notifications-when-your-aws-account-s-root-access-keys-are-used
- **API Management and Microservices** (06-22-2015 on www.windley.com) -http://www.windley.com//archives/2015/06/api_management_and_microservices.shtml
- **Google Offers Bounty for Android Bugs** (06-19-2015 on www.programmableweb.com) - http://www.programmableweb.com/news/google-offers-bounty-android-bugs/2015/06/19
- **Manage your API, without installing anything on your systems** (06-19-2015 on restlet.com) -

http://restlet.com/blog/2015/06/19/manage-your-api-without-installing-anything-on-your-systems/

- **You'll never guess which API documentation tools we use** (06-19-2015 on blog.lateral.io) - https://blog.lateral.io/2015/06/youll-never-guess-which-api-documentation-tools-we-use/
- **API Management Solutions Market Predicted to Quadruple by 2020** (06-17-2015 on www.programmableweb.com) -http://www.programmableweb.com/news/api-management-solutions-market-predicted-to-quadruple-2020/elsewhere-web/2015/06/17
- **On Air with the Google Apps Developer Team** (06-17-2015 on googleappsdeveloper.blogspot.com) - http://googleappsdeveloper.blogspot.com/2015/06/on-air-with-google-apps-developer-team.html
- **The Gumroad Small Product Lab: Accelerating Simple Ideas ·** (06-16-2015 on apievangelist.com) - http://apievangelist.com/2015/06/16/the-gumroad-small-product-lab-accelerating-simple-ideas/
- **The Gumroad Small Product Lab: Accelerating Simple Ideas ·** (06-16-2015 on apievangelist.com) - http://apievangelist.com/2015/06/16/the-gumroad-small-product-lab-accelerating-simple-ideas
- **A new Filepicker plugin….created by a Filepicker client** (06-16-2015 on www.filepicker.com) - https://www.filepicker.com/blog/fun/a-new-filepicker-plugin-created-by-a-filepicker-client/
- **Bitreserve opens API to developer community** (06-16-2015 on www.finextra.com) - http://www.finextra.com/news/announcement.aspx
- **Two ways to create your APISpark account** (06-16-2015 on restlet.com) - http://restlet.com/blog/2015/06/16/two-ways-to-create-your-apispark-account/
- **Intuit Developer Has a New Services Dashboard** (06-15-2015 on developer.intuit.com) - https://developer.intuit.com/blog/2015/06/15/intuit-developer-has-a-new-status-dashboard
- **5 Reasons Your API May be Ready for Retirement** (06-14-2015 on www.programmableweb.com) -

http://www.programmableweb.com/news/5-reasons-your-api-may-be-ready-retirement/elsewhere-web/2015/06/14

- **API KPI's – 4 Keys to Tracking a Successful Developer Program** (06-11-2015 on www.mashery.com) - http://www.mashery.com/blog/api-kpi-s-4-keys-tracking-successful-developer-program

Thanks for Tuning Into My Research!

Rembember -- You Can Find All Of This On Github (https://github.com/kinlane/api-management/), If You Want To Get Involved!

{"logo":"API Evangelist"}

@kinlane

@apievangelist

Make Sure And Share Your Public API Designs At The API Stack--Otherwise All Of This Won't Work!