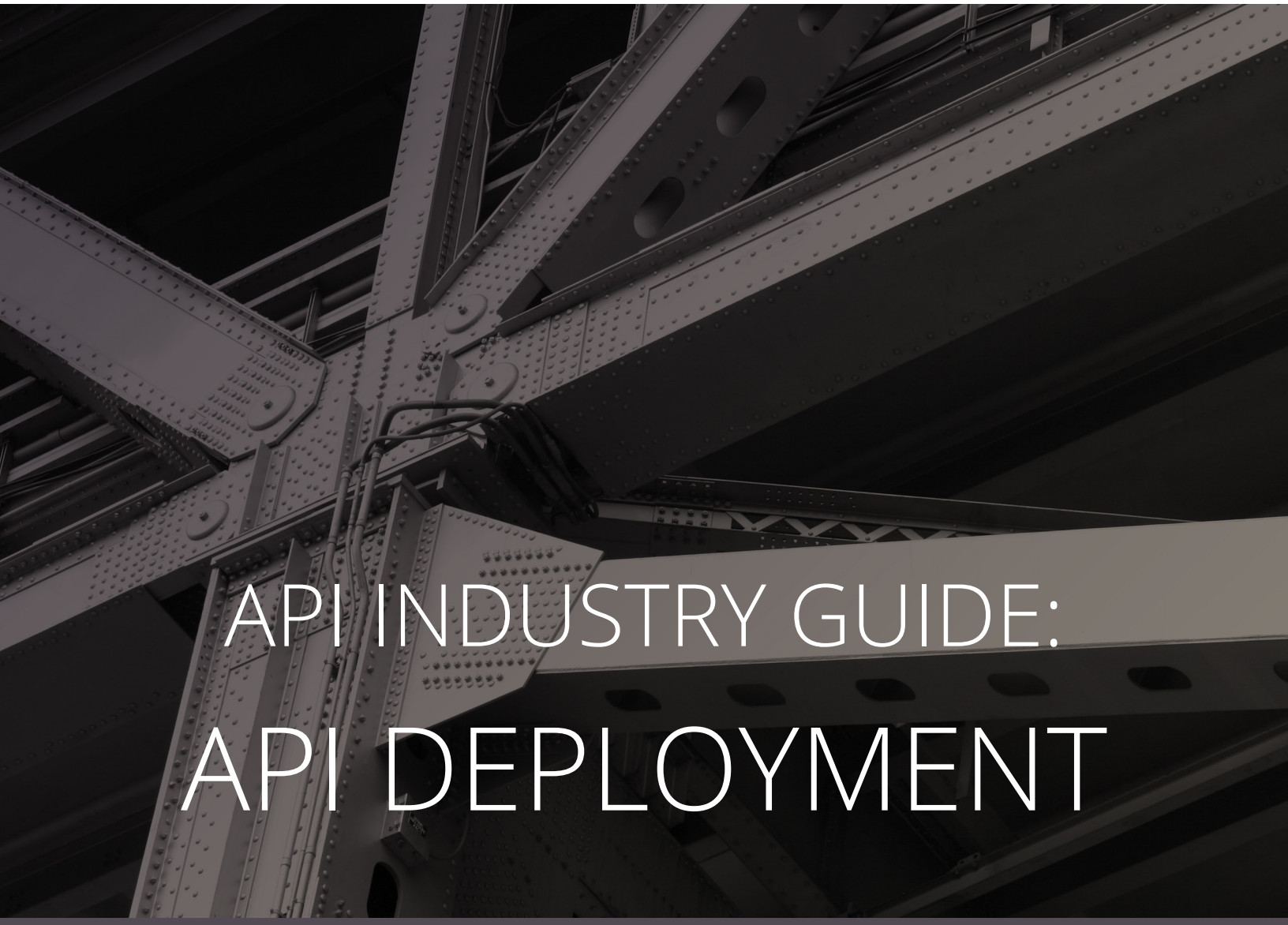
3scale by Red Hat | {"logo":"API Evangelist"}



API INDUSTRY GUIDE:
API DEPLOYMENT

JANUARY 2016 ■ BY KIN LANE, THE API EVANGELIST

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

AN OVERVIEW OF API DEPLOYMENT

There are as many approaches to API deployment as there are types of APIs. The why and how of API deployment varies widely, and until recently, deployment was a topic of conversation left to developers and IT groups. With the latest wave of growth in the world of web APIs, I've seen a more active conversation around how we deploy APIs both on-premises and in the cloud.

The history of API deployment has its roots in IT, as well as amongst web and mobile developers. Teams either deployed an API using a proxy or gateway (common for IT-led projects), or built their own from scratch, or—more likely—used an open source API framework for scaffolding (common in developer-led projects). Each of these approaches has its benefits, but some organizations or projects may not have the resources to cover the cost of a proper gateway, or to develop their own custom API, resulting in a wave of cloud and open source solutions which support the rapid deployment of APIs from databases, spreadsheets, and other sources.

The goal of this research guide is to help businesses be more aware of the high-level concepts surrounding API deployment before making an investment in tools or services. I hope to provide business and organizational leaders with the overall knowledge they need to understand the potentially different paths that are available for API deployment. Whether teams choose to bootstrap the initiative, or look for an API management service provider to assist, this project is meant to get readers of varying technical levels up to speed—with a healthy awareness—by providing the "view from 40,000ft."

We will walk you through the challenges of doing it yourself, how you can employ API frameworks, gateways, as well as a variety of cloud based, API deployment solutions. As the space is expanding I will also look at the evolving solutions for deploying APIs from spreadsheets, databases, and search indexes. There is a wealth of services, open tooling, and blueprints for API deployment to show you; almost too many to keep up with.

After reviewing this API deployment research, you'll understand the key factors involved in API deployment, and you'll be ready to think about API management.

THE COMMON BUILDING BLOCKS OF API DEPLOYMENT

There are a handful of common building blocks involved with API deployment. Depending on the resources you desire to open up access to, different building blocks will be needed.

These building blocks have been gathered from evaluating many public API providers, tools that have been developed by experienced API developers, and the services being offered and evolved by API service providers.

They are meant to be generic, modular concepts that will help on-board business or even technical folks to the expanding world of API deployment.

- **Framework** - There is no reason to hand-craft an API from scratch these days. There are numerous frameworks that are designed for rapidly deploying web APIs. Deploying APIs using a framework is only an option when you have the necessary technical and developer talent to be able to understand the setup of environment and follow the design patterns of each framework. It is best to select one of the common frameworks written in the preferred language of the available developer and IT resources. Frameworks can be used to deploy data APIs from CSVs and databases, content from documents, or custom code resources that allow access to more complex objects.

- **Proxy** - API proxies are commonplace for running an existing API interface through an intermediary which allows for translations, transformations and other added services on top of API. An API proxy does not deploy an API, but can take existing SOAP or XML-RPC resources and transform them into more common RESTful APIs with JSON formats. Proxies provide other functions such as service composition, rate limiting, filtering, and securing of API endpoints. API gateways are the preferred approach for the enterprise, and the companies that provide these services support larger API deployments.

- **Connector** - While proxies work in many situations, allowing APIs to be mediated and transformed into required interfaces, API connectors may be preferred in situations where data should not be routed through proxy machines. API connector solutions only connect to existing API implementations and are easily integrated with existing API frameworks as well as web servers like Nginx.

- **Hosting** - Hosting is all about where you are going to park your API. Usual deployments are on-premise within your company or data center, in a public cloud like Amazon Web Services or a hybrid of the two. Most of the existing service providers in the space support all types of hosting, but some companies, who have the required technical talent host their own API platforms. HTTP is the transport which modern web APIs put to use, sharing the same infrastructure as web sites, so hosting APIs does not take any additional skills or resources if you already have a website or application hosting environment.

- **Versioning** - There are many different approaches to managing versioning for web APIs. When embarking on API deployment you will have to make a decision about how each endpoint will be versioned and maintained. Many API service providers offer versioning solutions, often handled within the API URI or passed as an HTTP header. Versioning is an inevitable part of the API lifecycle, so it is better to integrate versioning by design as opposed to waiting until you are forced to make major changes in your API interface.

- **CSV to API** - Text files that contain comma separated values or CSVs are one of the quickest ways to convert existing data to an API. Each row of a CSV can be imported and converted to a record in a database, and easily generate a RESTful interface that represents the data stored within it. CSV to API can be very messy depending on the quality of the data in the CSV, but can be a quick way to breathe new life into old data. The easiest way to deal with CSV is to import directly into database, then generate an API from that database, but the process can be done at time of API creation.

- **Database to API** - Database to API is definitely the quickest way to generate an API. If you have valuable data; generally, it will reside in a Microsoft, MySQL, PostgreSQL or other common database platform. Connecting to a database and generate a CRUD (create, read, update and delete) API on an existing data make sense for a lot of reason. This is the quickest way to open up product catalogs, public directories, blogs, calendars or any other commonly stored data. APIs are rapidly replace database connections, when bundled with common API management techniques, APIs can allow for much more versatile and secure access that can be made public and shared outside the firewall.

- **Gateway** - Gateways expose internal systems and connect with external platforms. They are often used to proxy and mediate existing API deployments, but may also provide solutions for connecting to other internal systems like databases, FTP, messaging and other common resources. Many public APIs are exposed using frameworks, most enterprise APIs are deployed via API gateways--which support much larger deployments.

- **Scraping** - Harvesting data from an existing website, content or data source. While we all would like content and data sources to be machine readable, sometimes you have just get your hands dirty and scrape it. While I don't support scraping of content in all scenarios, and business sectors, in the right situations scraping can provide a perfectly acceptable content or data source for deploying an API.

- **Container** - The new virtualization movement, lead by Docker, and supported by Amazon, Google, Red Hat, Microsoft, and many more, is providing new ways to package up APIs, and deploy as small, modular, virtualized containers.

- **Github** - Github provides a simple but powerful way to support API deployment, allowing for publishing of a developer portal, documentation, code libraries, TOS, and all your supporting API business building blocks, that are necessary for API effort. At a minimum, Github should be used to manage public code libraries, and engage with API consumers using Github's social features.

THE DO-IT-YOURSELF APPROACH TO API DEPLOYMENT

When faced with the task of deploying an API, many groups size up the situation and then decide to do it themselves, incorrectly assuming that there's not much complexity involved. In practice, API deployment is a nuanced process, and there are many details to consider when crafting a consistent, stable API for a production environment. In short, it's not a good time to try re-inventing the wheel. This preference for a DIY approach is one reason why we see a diverse ecosystem of APIs today, many of which are bespoke projects, and why these APIs often don't work well together. Too many teams have ambitiously worked on individual projects without considering how their API will interact with others.

If your team doesn't have the budget for a high-end gateway solution, or for one of the emerging cloud-based solutions, there are many open source frameworks available to provide support and structure for the API deployment process--it just doesn't make sense to do this yourself.

For now, we'll explore the multitude of alternatives to the DIY approach, starting with the next best thing: using a common framework.

API DEPLOYMENT WITH FRAMEWORKS

For many involved in API deployment initiatives, using an existing open source API framework in your desired programming language is the way to go. If an organization has the available resources necessary to deploy and maintain a website, it can easily deploy and manage API infrastructure using any of the existing API frameworks.

An API framework can make basic deployment from a data source something a developer can tackle in minutes, rather than in days or weeks. Your team will have to select the framework that suits your development style, but most are intuitive and easy to adopt. It is likely that your developers, whether internal or third-party, are already using a framework to deploy web sites, web applications, or mobile applications, and that one of these frameworks can also be used to drive your API backend.

- **Apache Wink** - Apache Wink is a framework for building RESTful Web services. The Wink Client module is a Java-based framework that provides functionality for communicating with RESTful Web services. The framework is built on top of the JDK HttpURLConnection and adds essential features that facilitate the development of client applications.

  Website: https://wink.apache.org/

  Github: https://github.com/apache/wink/

- **Django REST** - The Django REST framework is a powerful and flexible toolkit that makes it easy to build Web APIs. It offers an attractive, web browse-able version of your API, and the option of returning raw JSON. The Django Rest Framework provides powerful model serialization and displays data using standard function based views.

  Website: http://www.django-rest-framework.org

- **OpenRasta** - OpenRasta is an open-source .NET framework for building everything web, from web sites to RESTful APIs. Written from the ground-up with testing in mind, OpenRasta helps you concentrate on writing clean, beautiful code. OpenRasta is a good choice for building scalable, high-performance, reliable web services.

  Website: http://openrasta.org/

  Github: https://github.com/openrasta

- **Restlet** - The Restlet API Platform enables developers and non-developers alike to design, create, run, and manage the APIs that provide access to any data or application. The Restlet Framework is the most widely-used open source solution for Java developers who want to create and use APIs. Restlet's APISpark product enables any organization to deploy an API quickly via a browser interface.

  Website: https://apispark.com/

  Twitter: https://twitter.com/apispark

  Github: https://github.com/apispark

Later, I list an even wider variety of open source frameworks that can make API deployment more efficient. In this section, however, I wanted to showcase a handful of high-profile frameworks that are backed by a company and/or well-organized community effort.

More frameworks will be added to this section over time, as these projects evolve and grow. It can be difficult to assess the usage and scope of any single API framework and its community. My goal is to just provide a variety of definitions to work with, which is something that pushes me to keep studying, and understanding each company, organization, or project—hopefully it will do the same for you.

USING API GATEWAYS FOR DEPLOYMENT

API gateways have been on the market for some time now, a carry-forward from the earlier evolution of APIs from the days of SOA (software oriented architecture). It can be tough to understand the features each gateway offers, and their comparative abilities to mediate between back-end resources and the end users of an API.

Gateways are a powerful way to address API deployment, and if this route makes sense for your company, I recommend starting a conversation with one or more of the providers listed below, and finding a company that understands your team's specific needs and goals. All of these API gateways have kept with the times, many evolving from their SOA roots, but now also understand the importance of delivering RESTful APIs, and providing a lightweight JSON structure for a variety of existing legacy, internal resources. Additionally, a handful of these gateways were born into this new web API realm, resulting in some very new views on what makes a gateway more usable.

Gateways are the best deployment option for larger businesses and teams that have more existing resources to expose. Deploying individual APIs with frameworks, and hand-rolling your own management tools, can quickly become overwhelming, so many of these providers have expanded their offerings, making their tools more compelling. Cloud providers such as Amazon and Google have been quickly delivering in this area as the overwhelming need for API deployment tools becomes more widespread.

All of this movement is resulting in a pretty seismic shift in area of API gateways in the past few years, some of which has been led by Amazon with their recent gateway release. Companies like JustAPIs are also pushing the boundaries and changing popular expectations about what an API gateway does. Just exactly what makes an API gateway, where or how you deploy it, and who can put it to use is rapidly being redefined, which makes this ongoing research very interesting.

- **3scale** - 3scale APIcast lets you deploy on a cloud API gateway service in just a few clicks, without modifications to backend code. APIcast is free to use and supports up to 50,000 API transactions per day, making it the perfect solution for low- or moderate-volume APIs. 3scale's unique hybrid architecture separates traffic control from the API management layer, providing greater flexibility and performance.

  Website: http://www.3scale.net/api-management/apicast/

  Twitter: https://twitter.com/3scale

  Github: https://github.com/3scale

- **Akana** - Monitor activity for specific partners or app developers and evaluate their impact on your business. Analyze which aspects of your APIs are being adopted, which devices or apps are being built against your APIs, study reasons for defections and incorporate insights from this analysis to improve your APIs and create best practices for developers.

  Website: http://www.akana.com/

  Twitter: https://twitter.com/akanainc

- **Amazon API Gateway** - 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

- **Apache Knox** - Apache Knox is a REST API Gateway for interacting with Hadoop clusters. The Knox Gateway provides a single access point for all REST interactions with Hadoop clusters, providing valuable functionality to aid in the control, integration, monitoring and automation of critical administrative and analytical needs of the enterprise.

  Website: https://knox.apache.org/

- **ApiAxle** - ApiAxle is a proxy that sits on your network, in front of your API(s) and manages rate limiting, authentication, and caching. It's fast, open and easy to configure. Unlike other cloud based services such as Mashery, it is intended to be installed within your LAN and be managed by your team.

  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. Apigee's API solutions help organizations use their enterprise data and services to create connected digital experiences for customers, partners and employees.

  Website: https://apigee.com

  Twitter: https://twitter.com/apigee

  Blog: https://blog.apigee.com/front

  Github: https://github.com/apigee

- **Axway** - Axway API Management offers 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/

- **Computer Associates** - Building on the foundation of its well-known SOA application gateway technology for exposing, securing and managing backend applications, network systems or infrastructure via APIs, CA Technologies has added critical mobile, cloud and REST composition features.

  Website: http://www.ca.com

  Twitter: https://twitter.com/cainc

  Blog: http://blogs.ca.com/

- **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

- **NGINX** - NGINX Plus is a trusted platform to manage and secure HTTP-based API traffic. Leveraged by leading API management platforms, NGINX Plus allows you to deliver API-based services with speed, reliability, scalability, and security.

  Website: https://www.nginx.com/solutions/api-gateway/

  Twitter: https://twitter.com/nginxorg

- **OpenLegacy** - OpenLegacy focuses on business needs instead of on replacing existing solutions. They offer a comprehensive integration platform to help companies to unlock the application services and information in mainframes, AS/400 and databases, automatically transforming them into mobile, web and cloud applications.

  Website: http://openlegacy.com/

  Twitter: https://twitter.com/openlegacy

  Blog: http://openlegacy.com/blog/

  Github: https://github.com/openlegacy

- **StrongLoop** - 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.

  Website: https://strongloop.com/node-js/api-gateway/

  Twitter: https://twitter.com/strongloop

  Github: https://github.com/strongloop

- **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. Extends 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

- **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. Supports hot-reloads so you can introduce new services without downtime.

  Website: http://tyk.io/

  Blog: http://tyk.io/blog/

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

  Website: http://www.wavemaker.com/

  Twitter: https://twitter.com/wavemaker

  Github: https://github.com/wavemaker

For this part of my research, the list of API gateway options feels, to me, like a catch-all bucket for anything that calls itself a gateway. With future iterations of this guide, I plan to further categorize, as I get to know each of the solutions better and discover other nuances.

Until recently, I didn't think of API gateways as an exciting topic of research; older gateway options did not provide developers with the tools they needed to truly turn resources into a high-quality API. But in more recent years, my opinion has changed, as I've watched gateway providers steadily build a more sophisticated suite of tools into gateways.

It's heartening to see that in particular, these forward-thinking changes provide improvements that benefit smaller API providers and enterprises alike.

THE EMERGING CLOUD API DEPLOYMENT PLATFORMS

Cloud computing has been a reality for some years now. While APIs were essential to the development of cloud computing itself, it is also a natural course for companies to also offer API deployment services that run exclusively in the cloud. A handful of companies have emerged in the last few years to provide API Deployment as a Service, connecting existing and new data sources, then generating web APIs, often complete with common API management services to boot.

Cloud-hosted API deployment from common data sources will make API deployment accessible to the masses, making the API deployment of numerous resources easier for developers, while also making API deployment a task which no longer requires a developer to complete.

- **Amazon API Gateway** - A fully-managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. 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

  Github: https://github.com/amazonwebservices

- **Apigility** - Separating presentation logic from raw data provides the flexibility to support multiple client form factors, and future-proofs apps to allow behind-the-scenes change without breaking user interfaces. With Apigility, you can make existing code API-enabled for more diverse use by customers, partners, and internal teams.

  Website: https://apigility.org/

  Twitter: https://twitter.com/apigility

  Github: https://github.com/zfcampus

- **Apitite** - Apitite is a web API creation and management service in the cloud which allows businesses to securely expose existing data to their users, employees, and partners automatically. Organizations can create web APIs by providing connection details to their data store, specifying which data to expose, and listing who should be granted access to the data, all through a web interface. Apitite provides the web hosting, security, and user access management for each API.

  Website: https://www.apitite.net/

  Twitter: https://twitter.com/apititeapi

  Blog: http://blog.apitite.net/

- **BinaryOps.io** - BinaryOps.io makes it easier to develop and deploy data APIs without building and hosting a dedicated database or re-writing aplication code.

  Website: http://www.binaryops.io/

  Twitter: https://twitter.com/binaryops

  Blog: http://www.binaryops.io/blog/

- **Deployd** - Deployd is an open source API design and deployment platform that allows developers to quickly design, customize, and deploy an API, with supporting application interface via a downloadable app and command line utilities. Deployd is a downloadable solution you can use on your desktop.

  Website: http://deployd.com/

  Twitter: https://twitter.com/deploydapp

  Github: https://github.com/deployd

- **DreamFactory** - DreamFactory publishes the DreamFactory Services Platform, an open source REST API platform for mobile application developers.

  Website: http://www.dreamfactory.com/

  Twitter: https://twitter.com/dfsoftwareinc

  Blog: http://blog.dreamfactory.com/

  Github: https://github.com/dreamfactorysoftware

- **Espresso Logic** - Espresso Logic provides a fast way to create RESTful APIs to multiple backend data sources with both read and write capabilities. Users can create an enterprise-class API complete with logic using reactive programming and fine-grain security, and API creation is declarative. Deliver backends—not just APIs—quickly and at minimal cost.

  Website: http://www.espressologic.com/

  Twitter: https://twitter.com/sqlrest

  Blog: http://www.espressologic.com/blog/

  Github: https://github.com/espressologiccafe

- **Google** - Create RESTful services and make them accessible to iOS, Android and Javascript clients. Automatically generate client libraries to make wiring up the frontend easy. Built-in features include denial-of-service protection, OAuth 2.0 support and client key management.

  Website: https://cloud.google.com/endpoints/

  Github: https://github.com/google

- **Instant API** - Instant API is Platform as a Service that makes it easy for developers to create, host, publish, and document APIs from existing data and services.

  Website: http://instantapi.co/

  Twitter: https://twitter.com/instantapi

- **Restlet** - The Restlet API Platform enables developers and non-developers alike to design, create, run, and manage the APIs that provide access to any data or application. The Restlet Framework is the most widely-used open source solution for Java developers who want to create and use APIs. Restlet's APISpark product enables any organization to deploy an API quickly via a browser interface.

  Website: https://apispark.com/

  Twitter: https://twitter.com/apispark

  Blog: http://blog.restlet.com/

  Github: https://github.com/apispark

- **Service Stack** - ServiceStack was developed with the mission of creating a best-practices services framework with an emphasis on simplicity and speed, reducing the effort in creating and maintaining resilient message-based SOA Services and rich web apps.

  Website: https://servicestack.net

  Twitter: https://twitter.com/servicestack

  Github: https://github.com/servicestack

- **StackHut** - StackHut turns code into live APIs, powered by Linux containers. Take any code, describe your config, and run StackHut deploy. Your functions and classes are automatically turned into APIs, accessible through a POST request.

  Website: https://stackhut.com

  Twitter: https://twitter.com/stackhut

  Github: https://github.com/stackhut

- **StrongLoop** - 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.

  Website: http://strongloop.com/

  Twitter: https://twitter.com/strongloop

  Github: https://github.com/strongloop

- **SwiftIQ** - SwiftIQ provides web-service application programming interface (API) infrastructure to facilitate data accessibility and predictive analytics. Swift Access is a backend platform which allows users to unify and secure disconnected data, while Swift Predictions allows users to apply adaptive, machine learning algorithms to make applications smarter.

  Website: http://www.swiftiq.com/

  Twitter: https://twitter.com/swiftiq

  Github: https://github.com/swiftiq

- **WaveMaker, Inc.** - WaveMaker Enterprise is an aPaaS software for rapid application delivery of enterprise custom apps. It offers out of the box support for security, web services integration and data modelling, a high performance cloud platform, and 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

- **Zatar** - Zatar is a product of Zebra Technologies which provides a standards-based approach to connectivity and control of devices along with open APIs to create apps, onboard devices and enable collaboration.

  Website: http://www.zatar.com/

  Twitter: https://twitter.com/zatar_iot

  Github: https://github.com/zatar-iot

Cloud API deployment is a pretty big bucket, so you can expect these descriptions and the list itself to change rather quickly. I believe many of these providers will continue to fine-tune their platforms and tools, pivoting until they find the right approach that solves problems for API providers at an appropriate price.

API DEPLOYMENT WITH FOCUS ON THE DATABASE

There are many reasons for deploying an API, but the most common reason I see is a desire to provide access to information stored in a database. While many API deployment providers allow for you to connect to a database, some providers have a stronger focus in this area, and I want to highlight them separately.

With the popularity of Hadoop, and emergence of Lambda from AWS, I expect to see the definition of what is "database", and what is API deployment from these databases, rapidly evolve. For now, here is what I'm tracking on in the area of API deployment using databases.

- **Apitite** - Apitite is a web API creation and management service in the cloud which allows businesses to securely expose existing data to their users, employees, and partners automatically. Organizations 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, all through a web interface. Apitite provides the web hosting, security, and user access management for each API.

  Website: https://www.apitite.net/

  Twitter: https://twitter.com/apititeapi

  Blog: http://blog.apitite.net/

- **CKAN** - CKAN is a powerful data management system that makes data accessible by providing tools to streamline publishing, sharing, finding and using data. CKAN is aimed at data publishers (government entities, companies, and organizations) who want to make their data open and available.

  Website: http://ckan.org/

  Blog: http://ckan.org/blog/

- **FluidDB** - Using a flexible underlying representation of information and a new model of control, FluidDB allows users and applications to work with information more naturally. That includes dynamically organizing, sharing, combining and augmenting information, and searching. It also allows users to choose exactly which information to share with whom, with separate controls for reading and writing.

  Website: http://fluidinfo.com/

  Twitter: https://twitter.com/fluidinfo

  Github: https://github.com/fluidinfo

- **Orchestrate** - Orchestrate provides a simple RESTful API service that eliminates the need for separate production databases for APIs. Orchestrate uses NoSQL databases to unify full-text search, social graph, activity feed, key/value document, and time-ordered event queries in a single interface, eliminating the operational and financial burden of deploying databases when building new applications.

  Website: http://orchestrate.io/

  Twitter: https://twitter.com/orchestrateio

  Github: https://github.com/orchestrate-io

- **SlashDB** - SlashDB connects to relational 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. By turning databases into online resources, content becomes accessible to authorized web, mobile, and enterprise applications for reading and writing under standard data formats.

  Website: http://www.slashdb.com/

  Twitter: http://twitter.com/slash_db

API deployment tools empower providers to go beyond simple resource-based API design, and allow them to evolve their designs and deploy much more meaningful endpoints, going beyond just the table and column names of source data. APIs provide a shortcut past the complexity of query language and directly access the raw resources beneath. Crafting simple, high value endpoints allows users to apply these resources in web, mobile, and other applications or integrate systems with ease.

API DEPLOYMENT USING SPREADSHEETS

Most of the world's business data is stored and manipulated within spreadsheets, so it makes sense that we would seek to deploy APIs from spreadsheets. I've seen a number of these types of API services providers come and go, but there is a current wave of them that are making a go of it.

These are the companies that make API deployment from spreadsheets something that anyone can do, with no developer skills required.

- **CKAN** - CKAN is a powerful data management system that makes data accessible by providing tools to streamline publishing, sharing, finding and using data. CKAN is aimed at data publishers (government entities, companies, and organizations) who want to make their data open and available.

  Website: http://ckan.org/
  Blog: http://ckan.org/blog/

- **Restlet APISpark** - APISpark is an cloud API platform that lets you create, host, manage and use web APIs. Using the Restlet Framework at its core, APISpark simplifies the web API experience, the time to market, and the overall cost to get started and to scale you APIs. Restlet is a web API platform vendor, pioneer of RESTful web APIs.

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

- **Sheetlabs** - Sheetlabs turns spreadsheets and other tabular information into well-documented APIs quickly and easily, making it easier to share product lists, internal information, and other useful data.

  Website: https://sheetlabs.com
  Twitter: https://twitter.com/sheetlabs

- **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 which allows government organizations to disseminate relevant information and data-driven services to a wide range of audiences.

  Website: http://socrata.com

  Twitter: https://twitter.com/socrata

  Blog: http://www.socrata.com/tech-blog/

  Github: https://github.com/socrata

In theory, you could launch an API from a spreadsheet by proxying the Google Sheets API, and the Office 365 API, but the services above make it much easier, and more importantly, provide a way to accomplish this goal in teams or organizations that are not able to dedicate developer resources to doing so. As we see the awareness of APIs grow among non-technical business users, I expect to see spreadsheet-to-API deployment solutions become more commonplace.

DEPLOYING APIS GENERATED FROM SEARCH INDEXES

Another source of quick, consistent, and potentially highly valuable APIs, is through indexing document and other data sources using search indexing solutions like Lucene. While these are not cloud solutions, they are potentially something a non-developer could do with a number of SaaS providers emerging that support Lucene and Elastic.

- **Apache** Solr - Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and more. Solr powers the search and navigation features of many of the world's largest internet sites.

  Website: http://lucene.apache.org/solr/

- **Elastic** - Elasticsearch offers an open source search and analytics engine available and make real-time data exploration available to anyone.

  Website: http://www.elasticsearch.org/

  Twitter: https://twitter.com/elasticsearch

  Blog: http://www.elasticsearch.org/blog/

  Github: https://github.com/elasticsearch

During my career I've seen a number of impressive API deployments in government organizations; frequently these APIs were based on raw data files that had been indexed using Elastic and deployed as fairly sophisticated APIs.

FUNCTIONAL API DEPLOYMENT USING CODE

The most common type of API is built to consume and deliver data, although there are many functional APIs that apply code and algorithms to make the magic happen.

One new trend I've observed is API deployment service providers which allow teams to upload or paste code, or connect with a Github repo. These deployment services then provide a wrapper around the provided code and launch an endpoint that puts the algorithm to work as an API.

This is an emerging niche within the broader API ecosystem, but here are some of the solutions for quickly deploying APIs from code.

- **Algorithmia** - A community built around state-of-the-art algorithm development. Users can create, share, and build on other algorithms and then instantly make them available as a web service.

  Website: https://algorithmia.com

  Twitter: https://twitter.com/algorithmia

  Blog: http://blog.algorithmia.com/

- **Apitite** - Apitite is a web API creation and management service in the cloud which allows businesses to securely expose existing data to their users, employees, and partners automatically. Organizations can create web APIs by providing connection details to their data store, specifying which data to expose, and listing who should be granted access to the data, through a web interface. Apitite provides the web hosting, security, and user access management for each API.

  Website: https://www.apitite.net/

  Twitter: https://twitter.com/apititeapi

  Blog: http://blog.apitite.net/

- **StackHut** - StackHut turns code into live APIs, powered by containers. Take any code - eg. Python, JS, native - describe your config, and run StackHut deploy. Your functions and classes are automatically turned into APIs, accessible through a POST request. Powered by Linux containers, it can now scale to meet any demand.

  Website: https://stackhut.com

  Twitter: https://twitter.com/stackhut

  Github: https://github.com/stackhut

If this area of API deployment can merge with the whole containerization world, we will really start seeing the server-side reality of the last 15 years melt away, replaced by a new world of API deployment solutions that allow you to paste any code language you want, and launch a standard web API interface on demand.

WEB SCRAPING AS A DATA SOURCE FOR APIS

Harvesting or scraping of content and data from other public web sources is something many do, but few talk about publicly. While scraping does infringe on copyright in some situations, in many other situations, it is quickly becoming a legitimate way to acquire content or data for use as an API.

In many cases, the owners of online content do not have the control, resources, or interest in making content available in a machine readable format. In these scenarios, for many developers, accessing this content for use in projects is as straightforward as creating a scrape script.

In response to this reality, we are seeing a new breed of API service providers emerge, who assist users in deploying APIs from data and content that has been liberated through harvesting or scraping. For the first time, several of these tools and services are mature and complete enough to merit a dedicated section.

- **Import.io** - Import.io turns the web into a database, releasing the vast potential of data trapped in websites. Allowing you to identify a website, select the data and treat it as a table in your database. Import.io effectively transforms the data into a row and column format. You can then add more websites to your data set, the same as adding more rows and query in real-time to access the data.

  Website: http://docs.import.io/

  Twitter: https://twitter.com/importio

  Blog: http://blog.import.io/

  Github: https://github.com/import-io

- **Kimono Labs** - Kimono is a way to turn websites into structured APIs from your browser in seconds. You don't need to write any code or install any software to extract data with Kimono; creating an API is as easy as adding a bookmarklet to your browser. The easiest way to use Kimono is to add our bookmarklet to your browser's bookmark bar.

  Website: https://www.kimonolabs.com/

  Twitter: https://twitter.com/kimonolabs

  Blog: http://blog.kimonolabs.com/

While I do have a separate set of research dedicated to scraping, which has other API-driven solutions for harvesting content from sites, the service providers listed above are specifically focused on deploying APIs from content that is scraped from websites.

EVERYONE IS DOING MICROSERVICES

While doing this research, we cannot forget about one of the hottest topics of conversation in the world of APIs right now—microservices. Whether you subscribe to the ideology or not, there is still a lot to learn from the movement as it pertains to API deployment.

Along with the wave of conversations, I'm seeing a new breed of API deployment providers emerge, speaking the microservice way. These are the ones I've investigated so far.

- **Dockpit** - Isolation for your microservice development process. Dockpit makes it easy to develop your microservice in isolation. It mocks the APIs you depend on and puts data stores, message queues and service registries in predictable states.

  Website: https://dockpit.io/

  Twitter: https://twitter.com/dockpit

  Github: https://github.com/dockpit

- **Seneca** - A microservices toolkit for Node.js. This toolkit lets you write clean code that you can scale without needing to refactor. Start with everything in one process, and split it all out onto multiple systems when you need to.

  Website: http://senecajs.org/

  Twitter: https://twitter.com/senecajs

  Github: https://github.com/rjrodger/seneca

Microservices is another area that I've only recently begun to focus on directly, and is still undergoing rapid evolution. In the next year or two, I'm curious to see if microservice-focused API deployment providers continue to pick up momentum, and I expect to make considerable updates to this section before I release the next edition of my API deployment research.

A WEALTH OF OPEN SOURCE TOOLS

There is an incredibly rich and diverse world of open source tools and resources available which support API deployment. In the earlier days of API adoption, the only thing available to developers were open source frameworks in a few popular languages.

Thanks to the massive growth in the API market, today's developers can choose from a long list of tools. Below, I've broken these down into similar groups, but I'm always interested to hear about new projects and ideas that will benefit the growing community of API developers, providers, and users.

Design Tools

- **Dockpit API Blueprint Template** (https://github.com/dockpit/pit-api-blueprint/) A Dockpit template for mocking Rest HTTP APIs using API Blueprint

Frameworks

- **Bottle** (http://bottlepy.org/docs/dev/) Bottle is a fast, simple and lightweight WSGI micro web-framework for Python. It is distributed as a single file module and has no dependencies other than the Python Standard Library.

- **CherryPy** (http://cherrypy.org/) CherryPy is a pythonic, object-oriented HTTP framework, that allows developers to build web applications in much the same way they would build any other object-oriented Python program. This results in smaller source code developed in less time. CherryPy is now more than six years old and it is has proven very fast and stable. It is being used in production by many sites, from the simplest ones to the most demanding.

- **DAVE** (https://github.com/evantahler/php-dave-api) DAVE is a minimalist, multi-node, transactional API framework written in PHP, which contains an end-to-end API test suite for TDD, a Task model, an Active Database Model, and a stand-alone development server (PHP) to get you started. These 4 methods make up the core functionality of many transactional web applications.The DAVE API aims to simplify and abstract may of the common tasks that these types of APIs require. DAVE contains an end-to-end API test suite for TDD, a Task model, an Active Database Model, and a stand-alone development server (written in just PHP) to get you started.

- **Django REST** (http://django-rest-framework.org/) Django REST is a lightweight REST framework for Django, that aims to make it easy to build well-connected, self-describing RESTful Web APIs. It automatically provides Django browse-able, self-documenting API and generates clean, simple, views for resources using Django's new class based views. Pluggable parsers, renderers, authentication and permissions makes it easy to customize.

- **Epiphany** (https://github.com/jmathai/epiphany#readme) The Epiphany framework is fast, easy, clean, and RESTful. The framework does not do a lot of magic under the hood. It is, by design, very simple and very powerful. The Epiphany framework never dictates how you should write or structure your application.

- **Falcon** (http://falconframework.org/) Falcon is a fast, minimalist Python web framework for building cloud APIs and app backends. The Falcon web framework encourages the REST architectural style, meaning (among other things) that you think in terms of resources and state transitions, which map to HTTP verbs.

- **Flask** (http://flask.pocoo.org/docs/) Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions. Flask's documentation is divided into different parts. They recommend that you get started with Installation and then head over to the Quickstart. There is also a more detailed tutorial that shows how to create a complete (albeit small) application with Flask. If you'd rather dive into the internals of Flask, check out the API documentation. Flask depends on two external libraries: the Jinja2 template engine and the Werkzeug WSGI toolkit.

- **Flight** (http://flightphp.com/) Flight is a fast, simple, extensible framework for PHP. Flight enables you to quickly and easily build RESTful web applications.

- **FRAPI** (http://getfrapi.com/) FRAPI is a new type of framework that embraces the standards and mindset of how the web is modeled. Instead of being a general purpose framework like the Zend Framework, Symfony, Cake, or Lithium. which are great for building web applications, FRAPI aims to remove the whole front-end layer complexity Handling REST calls can help users develop RESTful APIs that are easily scalable and highly performant.

- **Grape** (http://rdoc.info/github/intridea/grape) Grape is a REST-like API micro-framework for Ruby. Its designed to run on Rack or complement existing web application frameworks such as Rails and Sinatra by providing a simple DSL to easily develop RESTful APIs. It has built-in support for common conventions, including multiple formats, subdomain/prefix restriction, content negotiation, versioning and much more.

- **Limonade** (http://www.limonade-php.net/) Limonade is a PHP micro framework for rapid web development and prototyping. It's inspired by frameworks like Sinatra or Camping in Ruby, or Orbit in Lua. It aims to be simple, lightweight and extremly flexible.

- **Pylons** (http://www.pylonsproject.org/) The Pylons Project was founded to develop web application framework technology in Python. Rather than focusing on a single web framework, the Pylons Project will develop a collection of related technologies.

- **Recess** (http://www.recessframework.org/) Recess is a RESTful PHP framework that can be used by both beginner and seasoned developers. Recess is fast, light-weight, and has a very small footprint—ideal for LAMP development and drag-and-drop deployment to shared hosts. Recess is a modern framework that uses a loosely-coupled Model-View-Controller architecture designed and optimized specifically for PHP 5.

- **rest!** (https://github.com/silkapp/rest) rest is a set of packages used to write, document, and use RESTful applications. This API can then be run in different web frameworks, and allows you to automatically generate documentation, as well as client libraries for Haskell and Javascript.

- **Resteasy** (http://www.jboss.org/resteasy) Resteasy is a JBoss.org project aimed at providing productivity frameworks for developing client and server RESTful applications and services in Java. It is mainly a JAX-RS implementation but you'll find some other experimental code in the repository.

- **RestFixture** (https://github.com/smartrics/restfixture/wiki) A FitNesse fixture that allows developers, testers, and/or product owners to write test fixtures for REST API with simplicity in mind. The idea is to write tests that are self-documenting and easy to write and read, without the need to write Java code. The fixture allows test writers to express tests as actions (using any of the allowed HTTP methods) to operate on resource URIs and to express expectations about the content of the return code, headers and body.

- **Restify** (http://mcavage.github.com/node-restify/) Restify is a Node.js module built specifically to enable you to build correct REST web services.

- **RESTkit** (http://restkit.org/) RestKit is an Objective-C framework for iOS that aims to make interacting with RESTful web services simple, fast and fun. It combines a clean, simple HTTP request/response API with a powerful object mapping system that reduces the amount of code you need to write to get stuff done.

- **Restler** (http://luracast.com/products/restler/) A RESTful API server framework that is written in PHP that aids your mobile, web, and desktop applications. A framework, but with a difference – Restler can bend to meet your needs. Writing Server made easy and light. With the light weight, Restler makes writing a server as easy as writing it with just 3 PHP files. All public methods are automatically mapped to a URL. Just write class and add methods in PHP, and Restler takes care of everything else!

- **Restlet** (http://www.restlet.org/) Restlet is a lightweight, comprehensive, open source framework for the Java platform. Restlet is suitable for both server and client web applications. It supports major Internet transport, data format, and service description standards like HTTP and HTTPS, SMTP, XML, JSON, Atom, and WADL. A GWT port of the client-side library is also available.

- **RESTRack** (http://restrack.me/) A Model-View-Controller Framework. RESTRack follows the familiar MVC design pattern. Rack aims to provide a minimal API for connecting web servers and web frameworks. RESTRack leverages Rack to provide a minimal framework to create REST services. From the get go, RESTRack was designed to make it extremely easy to develop performant REST data services, making it perfect for data generation. Rich JavaScript frameworks such as ExtJS, jQuery UI, dojo, and native mobile applications would be well suited to for RESTRack serving as the data layer. The framework has a very small memory footprint, making it a great choice for cloud type architectures.

- **RESTx** (http://restx.mulesoft.org/) RESTx is a light-weight open-source platform for the creation of RESTful data access and integration resources and web services. It emphasizes simplicity, sane defaults and out-of-the-box usability. No complex configuration, no steep learning curve: You will be up and running in just 5 minutes. nRESTx is not your usual application framework and can simplify the creation of RESTful web services and resources.

- **Roar** (https://github.com/apotonick/roar) Roar is a framework for parsing and rendering REST documents. With Roar, REST representations are defined using a new concept called *representers*. Syntax and semantics are declared in Ruby modules that can be mixed into your domain models, following clean OOP patterns. Roar comes with built-in JSON, JSON::HAL and XML support, as well as client HTTP support, coercion, client-side caching, awesome hypermedia support and more. Roar is completely framework-agnostic and can be used in web kits like Rails, Webmachine, or Sinatra.

- **Seam REST** (http://seamframework.org/seam3/restmodule) Seam REST is a lightweight module that aims to provide additional integration with technologies within the Java EE platform as well as third party technologies. Seam REST is independent of CDI and JAX-RS implementations and thus fully portable between Java EE 6 environments.

- **Slim** (http://slimframework.com/) A simple yet powerful PHP 5 framework to create RESTful web applications. Slim lets you build a complete PHP web service with a single PHP file. Features include: RESTful routing, named routes, route passing, route redirects, route halting, custom views, and HTTP caching.

- **Spring Framework** (http://www.springsource.org/spring-framework) The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications on any kind of deployment platform.

- **Taffy** (http://atuttle.github.com/taffy/) Taffy is a ColdFusion framework that helps you build RESTful web services with very little boilerplate code, very little configuration, and to be honest, very little effort.

- **Tonic** (http://peej.github.com/tonic/) Tonic is an open source RESTful PHP library for web application development. Tonic helps you develop web applications that embrace the way the Web really works, enabling your applications to scale, extend and work with other systems easily.

- **Wave Framework** (http://www.waveframework.com/) Wave is a PHP micro-framework that is built loosely on model-view-control architecture and factory method design pattern. It is made for web services, websites and info-systems and is built around a native API architecture, with smart image and resource management. Wave is a compact framework that does not include optional libraries, has a very small footprint and is developed keeping lightweight speed and optimizations in mind.

- **web py** (http://webpy.org/) web.py is a web framework for Python that is as simple as it is powerful. web.py is in the public domain; you can use it for whatever purpose with absolutely no restrictions.

- **Zend** (http://framework.zend.com) Zend_Rest_Server is intended as a fully-featured REST server. To call a Zend_Rest_Server service, you must supply a GET and POST methods, with a value that is the method you wish to call. You can then follow that up with any number of arguments using either the name of the argument or using arg following by the numeric position of the argument. When returning values, you can return a custom status, you may return an array with each status.

The number of open source API frameworks has steadily grown over the last couple of years, while open source also takes root in other areas like design, management, and integration. As the API gateway world expands, I will be taking another look at the open tooling available, and refresh this list with anything relevant.

TAKING YOU FROM API DEPLOYMENT TO API MANAGEMENT

I have broken out API design and deployment into individual research findings, separate from API management. All of these concepts are intertwined and overlap is inevitable, but it is important to understand the overview of API deployment, from framework to gateway, and where it dovetails with API design and management.

API deployment was born out of the enterprise SOA world, and API gateways have the longest history of use in API deployment. But over the last 10 years a wealth of frameworks and scrappier approaches to deploy APIs build on HTTP has evolved considerably.

After 10 years of evolving API frameworks and RESTful approaches, a new breed of API providers have emerged to support the next generation of API, which are driving mobile apps as well as the Internet of Things. More importantly, these tools have also helped make API deployment something anyone can do--not just technical and developer teams.

The API Evangelist network is focused on educating the masses around the potential of APIs, helping everyone to understand the value of API deployment, even if you won't be getting your hands dirty with the actual nuts and bolts of it..

You may have a wealth of data locked up in databases, but lack the internal resources to design, develop and deploy your APIs. Five years ago, APIs were a purely technical initiative, born out of IT and developer groups. Today, anyone with a little curiosity, a business objective, and a willingness to dig in can understand what is possible, and by being involved, can help shape the future of the API industry.

Thanks for tuning in to my research!

Remember – you can find all of this on GitHub if you want to get involved!

{"logo":"API Evangelist"}

@kinlane | @apievangelist