

GIGAOM RESEARCH

A field guide to web APIs

Kin Lane

October 23, 2013

This research report is underwritten by Restlet.

TABLE OF CONTENTS

Executive summary3

What are APIs used for?4

Why are web APIs different?5

History of web APIs7

What technology goes into an API?.....8

Deploying your web API11

Established practices for managing APIs12

Marketing and API evangelism14

The Future of Web APIs17

Cloud Trends18

Key takeaways19

About Kin Lane20

About Gigaom Research20

Executive summary

A new breed of web API has emerged, delivering a vision of a lightweight, low-cost approach to connect devices and allowing applications to exchange data efficiently.

This research report is a field guide for web API providers, developers, and even nondevelopers. Among its key points:

- Previously APIs required custom technology, protocols, and often costly implementations, but web APIs build on existing technology, specifically the HTTP protocol that the world wide web employs to deliver content worldwide — with no need to invest in new technology.
- Simplicity rules. The web API providers that simplify interfaces and encourage adoption (even to a nondeveloper audience) are emerging as the leaders.
- The purpose of web APIs is to deliver valuable, scalable, and distributed resources across the internet. This requires four ingredients: commerce, social, the cloud, and mobile.
- Application-developer onboarding requires clear steps. App developers should be able to register, authenticate, and access documentation, code samples, support, and any other resources essential to API integration. And they need to do it easily.
- Opportunities for web APIs are emerging in the internet of things (IoT) — internet connectivity applied to everyday objects — in automobiles, homes, and building connections as well as in health and fitness and in manufacturing with 3D printing.

What are APIs used for?

The many shapes and sizes of this new wave of web APIs allow them to be used in an increasing number of ways. APIs in their simplest form allow the transmission of data, much like you'd share data within a spreadsheet by emailing it to one or many co-workers. But in this case, it's for applications. In more-advanced approaches you see APIs deployed to deliver video, audio, and telephony services via smartphones, televisions, or internet-enabled kiosks.

Open data

The most popular approach to APIs is the open-data movement. Using web APIs, data owners can easily expose information from data silos, allowing public or private access to large and small datasets. This read and sometimes write access to data is a favorite among developers, providing them free or low-cost access to data that they don't have to gather, curate, and organize. Open data has picked up significant momentum within city, state, and federal governments. It provides access to public data assets ranging from census data to U.S. postal service zip codes.

Websites

APIs provide access to data and content, enabling a new, distributed approach to website development. Rather than hand coding websites and their content, developers can pull data and content via APIs, then mashup and publish across web pages and applications they author. In 2003 much of the content on a web page likely came from a database source, but in 2013 a single web page will likely show content, advertising, links, and widgets that could be pulling the information from multiple API-driven sources. APIs have allowed technology giants like Amazon and eBay to drive content on their own web properties as well as establish armies of affiliate and partner electronic storefronts that are updated through centralized API sources.

Mobile

APIs have been a driving force in the expansion of the world wide web, but the introduction of the iPhone in 2007 and the resulting developer-and-application ecosystem proved that the power of APIs is not just to deliver data and content to websites. APIs could also provide the resources to drive the next generation of applications that would be consumed via a wave of smartphones and tablets. Developers discovered that APIs were a perfect way to deliver the bits and bytes needed to craft the mobile-user experience but also sync users' increasingly online experience via what was becoming known as the cloud.

Automobiles

As our lives become more internet-connected via our mobile devices, a parallel connectivity would naturally emerge via the other device that plays a leading role in our daily lives: the automobile. Major automakers like Ford, GM, and even newcomer Tesla have acknowledged the potential of internet-enabled vehicles and the efficiency of using web APIs to deliver rich data, content, and media to drivers and passengers. APIs provide a real-time, geo-aware experience in vehicles that allows users to navigate, listen to internet-streaming music, and even interact with the road, signs, and businesses around them via an emerging ecosystem of API-driven networks, services, and resources.

Homes and buildings

Originally the internet reached the world via desktop computers in homes and businesses. Migrating beyond those spaces via mobile devices, the internet is experiencing renewed growth. The internet is penetrating homes and offices via a wealth of internet-connected devices, known as the internet of things (IoT). The internet is being embedded in every aspect of technology in ways unimaginable just 10 years ago.

Manufacturers are making thermostats, lighting, climate control, energy, entertainment, security, and every other aspect of residential and commercial infrastructure internet-enabled. IoT providers are taking advantage of existing approaches to web API, developer ecosystems, and the availability of smartphone devices to give the masses control over their homes and work environments in new and exciting ways.

Why are web APIs different?

Even with its 30-year history, something about the current approach of employing APIs over the world wide web is different. Web APIs resemble their earlier incarnations, but they have escaped those earlier technology overlords that looked to keep them strictly as a programmatic interface. The latest generation of mobile and web developers is accessing resources in ways that previous developers couldn't imagine without the introduction of the internet, the cloud, and mobile devices.

They build on existing web architecture

Earlier APIs required custom technology, protocols, and often costly implementations to deploy, manage, and operate. Web APIs build on existing web technology, specifically the HTTP protocol that the world

wide web employs to deliver content around the globe. This reuse of existing architecture enables cost-effective and rapid deployment of API-driven platforms: No new technology investment is needed.

A particular methodology call REST, defined around 2000, provides a blueprint that is used as the current definition of web APIs. This model is being iterated upon with new definitions like Hypermedia and using the latest HTML5 standards like WebSocket. By using existing web architecture to deploy APIs, businesses can focus on delivering the best resources to web and mobile applications, alongside existing investments they have made for earlier websites and applications.

Intuitive resources

APIs are meant to enable software applications to talk with one another, but ultimately humans must build these applications and define their interactions. The recent explosion in the adoption of web APIs is because the latest generation of APIs can define data and resources in intuitive ways that humans can read and understand. Web users understand the concept of the URL and URI even without fully understanding the technology behind it. Web APIs build on the same awareness to provide access to resources and data, creating a more intuitive approach to sharing and accessing information. In online ecosystems where developers have a wide range of choices, they are more inclined to select API resources that are intuitive and don't require huge amounts of time and investment to understand.

Simplicity rules

In a world swimming in a sea of technology solutions, simplicity rules. Earlier incarnations of APIs were technical, often bloated interfaces that only the savviest of technologists could use. Web APIs, employing proven web patterns, are delivering extremely simple, lightweight approaches to data and resource sharing. Some API providers still employ complex technical interfaces, but web API providers, who go the extra mile to simplify interfaces and reduce friction in adoption, are emerging as the leaders in this fast-paced online environment.

Easy to understand for developers or even nondevelopers

Developers are quickly becoming web and mobile literate, understanding software development in terms of the internet. Web APIs provide simple and intuitive access to resources developers need to solve the challenges they face. Web APIs use intuitive URIs to access resources in ways that any web-literate user can take advantage of, making API-driven resources accessible to a wider developer audience and even a nondeveloper audience. They bring valuable resources and assets closer to the owners seeking solutions. APIs provide data and resources up to multiple channels including web, mobile, and other devices. They

also provide easy access via known tools like spreadsheets and embeddable widgets, expanding the audience of API consumers beyond developer communities.

Self-service resources

A common approach to delivering API resources is employing a self-service environment. While resources can still be secured as desired, APIs can be accessed 24/7 via a dedicated portal, allowing consumers to register and gain access without the need to talk to sales or other onboarding teams. When developing web and mobile applications, developers are finding success by quickly finding the resources they need to make their apps a reality — all without the bottleneck of requesting access or purchasing licenses. Self-service APIs have become the norm. Developers expect that resources ranging from data to compute to storage will be available at any time and in any volume they are willing to pay for.

History of web APIs

In 2013 the success of web APIs is not just technical. We can learn about many other, less-obvious aspects of web APIs by studying the past and looking at why some of the pioneers of web APIs were successful.

Commerce

As soon as the first dot-com bubble burst, platforms were looking for innovative ways to syndicate products across ecommerce websites. Web APIs, built on the backs of existing HTTP infrastructure, proved the right tool for the job. A handful of tech pioneers stepped up to define the earliest uses of web APIs for sales and commerce management, kicking off a 10-year evolution that we can now view as the early history of web APIs.

Social

As API-driven commerce platforms were still getting their footing and understanding the best way to use APIs, a new breed of technology platforms emerged in the areas of content and messaging on the web in a way that was user-centric and socially enabling. Publishing user-generated content and sharing web links, photos, and other media via APIs emerged with the birth of new social platforms between 2003 and 2006. These new API-driven social platforms would take technology to new heights and ensure that applications

from here on would always contain essential social features that were defined via their platform APIs. This proved that social was the critical ingredient the API industry needed if it was to mature.

Cloud computing

As APIs were generating social buzz across the internet, Amazon saw the potential for a RESTful approach to business, internalized. It gave birth to an API approach that was more than just ecommerce. It would reinvent the way we compute.

Amazon transformed the way we think about building applications on the web. What we now know as cloud computing changed everything and made the mobile, tablet, sensor, and other emerging API-driven concepts a reality.

Mobile

With the introduction of iPhone and Android smartphones, APIs would evolve from powering ecommerce, social, and the cloud to delivering valuable resources to the mobile phones that would rapidly become commonplace. APIs make valuable resources modular, portable, and distributed — a perfect channel for developing mobile and tablet applications. A wealth of mobile applications would emerge, focusing on delivering features that were unique to a mobile audience, pushing API usage to new levels.

Understanding history is critical to understanding where we are going, and there are many important lessons to be learned from the approach of web API pioneers. If early technologists had had their way, APIs would have been solely successful in the period of commerce. But with the radical innovation of a handful of companies, we now understand that APIs needed several other essential ingredients to succeed: commerce, social, cloud, and mobile.

Web APIs are about delivering valuable, meaningful, scalable, and distributed resources across the world wide web, and these ingredients were necessary to fully deliver on this vision.

What technology goes into an API?

APIs are driven by a set of specific technologies, making them easily understood by a wide variety of developers. A focus on simplicity means that APIs can work with any common programming language and be understood by any programmer, even one with little or no training in API technology.

REST

The most popular approach to delivering web APIs is Representational State Transfer (REST). This approach to API design takes advantage of the same internet mechanisms used to view regular web pages, so it has the advantage of faster implementations and is easier for developers to understand and put to use.

REST APIs allow you to take information and functionality that is already available on your website and then quickly make it available through a programmatic API, so that both web and mobile applications can reuse it, dramatically extending your company's reach over new channels, all without much additional work.

JSON

JavaScript Object Notation (JSON) is a way for programs to exchange information. APIs are a way for programs to communicate, but since they don't have voices, they need a way to describe the data and information they that is being exchanged. JSON uses brackets, quotes, colons, and commas to separate data, giving the information meaningful structure so that computers can differentiate between a first name and last name or any other information that potentially describes data. JSON has become one of the preferred ways for programmers to enable API communication. It provides a lightweight, simple way to exchange data across the internet while maintaining the structure and meaning of that data.

Security

Security is the No. 1 concern around web API deployment. Companies making valuable data and assets accessible over the internet require solid approaches to securing these resources. Over the past 10 years, the modern web API movement has developed proven techniques for securing the growing number of API resources.

Keys

One of the most common ways to secure API access for applications is to use what is called a key. This is often just a combination of alpha and numeric characters that each developer can obtain via a simple API-registration process. Each API consumer must include a key with each call to an API, allowing API providers to audit usage and issue or revoke keys as they deem necessary.

Basic auth

Basic auth is a way to provide credentials in the form of a username and password. Because basic auth is integrated into HTTP protocol, it is the easiest way for users to authenticate with a web API. Basic auth is easily put to use. However, if SSL is not used, the username and password are passed in plain text and can be easily intercepted over the open internet. Basic auth is perfectly suited for APIs that are intended for a wider audience but do not give access to overly sensitive information.

Open authorization

Open authorization (OAuth) is an open standard for authorization that allows users to share their private resources with another application without having to hand out their private credentials (typically their username and password). Instead of giving out private credentials, OAuth allows users to hand out unique tokens that give access to a specific site set of resources. OAuth is a service that is complementary to, but distinct from, OpenID, which is about defining identity. OAuth is about defining identity while also establishing which resources that person has access to. This is an excellent option for API security.

Securing APIs is essential to ensure proper access to resources, but web APIs are not just about one-way access to resources. There is also proven technology for pushing data back out to API consumers, making API operations truly a two-way street.

Webhooks

Webhooks are a form of push notifications that can be triggered by a specific action. When triggered, they push information to an external website address. Webhooks allow developers to choose the action, URL, and fields associated with the webhook push. Once a webhook is triggered, it passes all associated information to the location where a developer can process it. Webhooks are a great way to reduce constant polling on an API, because they push data to API developers only when the required action is triggered. Webhooks make the API a two-way street, allowing developers to not only pull data from API platforms but also receive data in real time as events occur.

REST, JSON, OAuth, and Webhooks have become the preferred technology for both API providers and API consumers, because they stick with the core principles of simplicity, security, making data and resources accessible, and easily integrating into web and mobile applications. This suite of web API technologies was not designated by a single standards body or by a single company. It has been established over the past 13 years through the best practices of existing, successful API providers that are meeting the demands of developers.

Deploying your web API

Historically companies have deployed APIs either with internal development teams or by purchasing software solutions that generate APIs from existing systems. In the past couple of years, as the demand for web APIs has grown, a new breed of services and cloud platforms has emerged to help companies, organizations, and even government organizations deploy APIs. The three distinct approaches to API deployment are do-it-yourself (DIY), cloud, and enterprise gateways.

Do-it-yourself approaches

An existing open-source API framework in the desired programming language is the best approach for many companies tackling API deployment. A company with resources to deploy and maintain websites can easily research and deploy APIs using the growing variety of available API frameworks such as Restlet or Slim Framework. Web APIs can vary widely in their implementation, but some common patterns have emerged, resulting in a nice selection of open API frameworks. A developer with an API framework can tackle basic API deployment from a data source in minutes rather than hours or days, following common patterns API providers have established and developers have grown used to.

Cloud solutions

Cloud computing has become an IT powerhouse over six years of solid growth. While APIs were essential to the development of cloud computing itself, some new models of API deployment services run exclusively in the cloud. A handful of solutions such as APISpark, SlashDB, and Emergent One have emerged in the past year to provide API deployment as a service. They connect to existing or new data sources and then generate web APIs complete with a portal and management tools. Cloud API deployment from common data sources is making API deployment accessible to the masses, enabling API deployment of numerous resources easier for developers while making API deployment something even a nondeveloper can understand.

Enterprise gateways

API gateways are a software solution for API deployment and a viable way SMB or enterprise organizations can effectively deploy web APIs from existing systems. API gateways have kept with the times. After evolving from their SOA roots, they understand the importance of delivering web APIs, providing lightweight JSON from a variety of existing legacy internal resources. API gateways allow companies to maintain legacy systems that were built using a variety of platforms while evolving toward a

web API strategy that will keep pace with modern practice and meet the demands of web and mobile app developers. They are available from providers ranging from Intel to Apigee.

Established practices for managing APIs

Practices for managing APIs grew out of early enterprise roots, just as the technology for deploying APIs did. But being defined by the early internet startups such as Salesforce.com, Amazon, and eBay and iterated upon by newcomers such as Twilio, SendGrid, and Foursquare, web APIs evolved beyond the enterprise. By about 2006 a common approach emerged for managing web APIs via developer portals. By 2013 more than 20 API-management service providers were providing a wide range of services from access management and analytics to complete API gateway services that help companies deploy and manage hundreds of APIs from internal resources.

API management usually begins with a portal site, where API providers host the operations of one or many API resources, including not just the management of the technical API interfaces but also the blog, forums, documentation, support desk, and other critical building blocks of API operations. API management requires ongoing, iterative research and development around a company's data and resources. API management will define the next phase of how business development will occur.

Self-service

One of the core philosophies of the web API movement is that APIs should be self-service. Even if an API is not meant to be public, there are plenty of ways to provide self-service access while still securing the API. One is using invitation codes or limited-entry service tiers that limit access when developers first register. Innovation around an API occurs around the clock; developers don't want to wait for access. Allowing developers to register using their existing social network accounts like Twitter, Facebook, and Google and specifically GitHub (which is the most logical of social networks for developers to use when registering for an API) reduces friction during registration.

Getting started

API developer onboarding requires simple, clear steps that help developers register, authenticate, and access documentation, code samples, support, and any other resources essential to API integration. Developers should be able to access a quality guide to getting started from the API landing page. A

getting-started page should provide what a developer needs to get started with as few steps as possible. This will make developer onboarding frictionless and increase the chances of a successful integration.

Documentation

Complete, easy-to-understand, and up-to-date API documentation is the doorway to API integration for developers, but it takes a great deal of work from more than one person on an API-development team. After the API developers complete their draft, someone outside the development team should review it. Like the API itself, documentation should be simple and intuitive. It should deliver value and not slow a developer down.

Code samples

A common item in the most successful APIs is working code samples in all of the top programming languages. Documentation will describe how to use an API, and code samples will speak in the specific language of developers. API providers deliver a wide variety of code samples in popular web languages like PHP, Python, and Ruby as well as lesser languages like ColdFusion or Perl, and they can often take into consideration the needs of enterprise developers with Java, C#, and even Visual Basic. Developers may not speak HTTP and REST, but they will speak one of these programming languages fluently. Code samples make an API as accessible as possible and ensure reaching the widest possible audience.

Support and feedback loops

API programs must also provide easy access to the support resources developers need when they get stuck. Support starts with self-service materials like forums, FAQs, and online communities like Stack Exchange and Quora. API providers should also provide direct-support options like regularly defined office hours, email, phone, or ticketing system support, which will give developers a direct line for getting their needs met. With both self-service and direct-support options, API providers can successfully establish feedback loops that take common questions and the needs of consumers and integrate them into the API road map and self-service support materials. Feedback loops will make support an ongoing, organic flow of information that will not only help API integrators be successful but also make the APIs better themselves.

The legal aspect

Added to the technical and business considerations of exposing data and resources via APIs are the legal considerations of how these assets will be accessed and used. API providers craft terms of use to establish

how developers can access and use resources. The terms of use also address privacy protections, which are essential for developers to understand.

A second major concern for companies and organizations deploying APIs is losing control over branding. Establishing clear and effective branding guidelines as part of API operations is an essential part of the legal building blocks for any API platform.

Developer dashboard

Developers make a considerable investment in time and money when they integrate with an API. API platforms usually provide developers with a dashboard to manage their API integration and operations, using common building blocks such as:

- Usage logs and analytics
- Application keys for potentially multiple integrations
- Basic settings for management, reset password, and billing history

In 2013 developers depend on multiple API resources, potentially from several separate platforms, so they need a single dashboard for all the important aspects of their usage and integration.

Marketing and API evangelism

With the technical, business, and legal side of API operations in place, each API provider should consider a marketing strategy. The best way to spread the word about the value an API delivers is widely known as evangelism in the space.

Goals

Establishing clear goals for an API is often overlooked and could cause an API initiative to wander and not get traction. Common goals for API programs are:

- Increasing overall awareness of the program
- Growing the number of new users
- Encouraging existing users to increase usage of API resources

Program goals shouldn't be limited to these common target areas, and API providers should consider other benefits, including overall brand awareness of the company and its products or services.

User engagement

One of the most important vehicles in API marketing and evangelism is existing consumers. When you already have someone's attention, engage, engage, and engage.

1. When new users sign up, engage them right away via the channel that makes most sense, in a way that doesn't feel like sales pressure.
2. Engage users regularly throughout their integration life cycle.
3. Even when users are actively integrated, let them know what is going on with the platform; encourage them to be word-of-mouth vehicles for your overall marketing initiative.

Email is probably the default channel for every new API registration, but work to discover other viable social channels, such as Twitter, LinkedIn, and GitHub, which will act as amplification for all user engagement and will help drive overall growth and adoption.

Blogging

Storytelling is the single most important aspect of API marketing and evangelism. The most common approach is to blog projects that are being executed by third-party developers and craft them into case studies and relevant stories for new and existing API consumers. A healthy syndication strategy that supports each story includes sharing it on top social networks as well as on social bookmarking platforms like Reddit and Hacker News.

Landscape analysis

Marketing APIs is about participating in a larger online community of developers, providing value and resources that make sense in a larger world of web and mobile application development. Participating in this new application economy requires awareness of the overall industry and technology trends, which, in turn, requires an ongoing analysis of the landscape of competitors, developers, and technology providers. This aspect of API marketing and evangelism is most akin to existing social media and online marketing practices that target specific keywords, topics, and markets and regularly monitor other blogs, social media users, forums, and search engine results for targeted areas. Landscape analysis provides the necessary intelligence to adjust overall API marketing strategy, keeping efforts in line with the larger

online industry and developer community and making sure API resources actually deliver value and provide real-world solutions to technology problems.

GitHub

One community of online developers stands out from the world of social networks: GitHub. This fast-growing social-coding platform is the go-to place for developers from all disciplines and is the most important network where API providers can engage. API platforms are using GitHub in myriad ways, but the common approaches include publishing code samples and documentation and providing support via GitHub issue management and wiki tools. GitHub provides a relationship-management platform for API providers and consumers. Allowing real-time engagement around code and API interfaces, it plays a central role in any API marketing and evangelism campaign.

Social

An active social media presence is a leading way to generate new API consumers and engage active API users, building overall momentum and awareness around an API platform. Twitter, Facebook, LinkedIn, and Google Plus are central to any business social media strategy. They deliver value to API evangelism efforts, but they also work as the central channels of support and feedback loops that are clear signals of any healthy API program. New users will judge an API platform based on its social activity as well as the response time of platform owners. Social activity can be a major signal of whether developers can depend on a resource for their applications and business.

Events

A significant portion of API evangelism will occur online, but marketing an API also requires going offline and engaging with developers at events. Conferences are a clear way to exhibit, speak, and reach developer audiences, but new, smaller events can provide a more intimate way of engaging with developer communities. Every town across the country hosts meetups on topics ranging from mobile development to Java user groups. These provide an excellent channel for reaching developers. Additionally, hackathons have emerged in the past few years. API owners can bring their API resources to weekend-long hackathons where developers can plan and build applications in a group setting, showing the value delivered via API resources in real time. Conferences, meetups, and hackathons build trusted relationships with developers and are an essential part of any API marketing and evangelism program.

No single formula fits every API marketing and evangelism campaign. Even though API pioneers and leaders have established numerous proven tactics, ultimately each API provider must understand which

tactics should be assembled into an API marketing strategy. Each strategy should consist of executing within short, agile cycles that allow providers to find what works and quickly discard approaches that do not. API marketing is about trying lots of different things and failing often but ultimately finding lots of little successful tactics that can be executed as part of a conscious overall iterative strategy.

The future of web APIs

The world of web APIs is evolving quickly. Emerging trends are rapidly shaping how APIs are provided and consumed. API usage is evolving, further connecting the data and resources we need to build the next generation of web and mobile apps.

API aggregation

Developers are depending on more APIs to drive web and mobile applications. Even with the simplicity of individual web APIs, developing an application around multiple APIs can be costly and time consuming. In the course of developing a common mobile application a developer may end up using 10 to 15 APIs — for storage, images, email, SMS, voice, and so on — increasing development costs with each addition. New approaches to the aggregation of multiple APIs into a single interface or stack for developers to use are addressing this problem by standardizing interfaces and objects across potentially disparate and numerous API platforms.

Real-time APIs

Technology has changed the pace of how markets operate. Real-time information has become the new default. We get push notifications when our friends post new tweets or Facebook wall posts. Our news, stock, and sports information streams to us in real time. With the internet available in anyone's pocket, the opportunity for real-time interaction with what is important to everyone exponentially increases. Real-time approaches to deploying APIs use technological building blocks like PubSubHubbub, WebSocket, and Webhooks, and they can be as simple as sending an SMS or email to signal an update.

Backend as a Service (BaaS)

Backend as a Service (BaaS), also known as Mobile Backend as a Service (MBaaS), is an emerging trend in mobile-application development. With the growth in the number of mobile devices, BaaS providers are emerging to meet demands by efficiently building, deploying, and managing the mobile app life cycle. This new breed of BaaS services is provided via custom software development kits (SDK) and application

programming interfaces (APIs), resulting in a global market that is estimated to grow from \$216.5 million in 2012 to \$7.7 billion in 2017, according to a report published by MarketsandMarkets.

Automation

Businesses and individuals moving more of their lives into the cloud require more automation of the tasks that keep these disparate worlds connected. Users don't just need to sync the individual information silos. They need to automate the transfer of information and resources. Many of the common cloud platforms have RSS and APIs that allow a new type of automation platform to emerge, providing new ways to extract, transform, and publish information and data across the web and devices.

Voice

We remain a verbal society. Even though the keyboard and mouse are the default input for the online world, many people now have an internet-connected device in their pockets, so the opportunity for voice search and other audio-based online interactions has grown significantly. Voice is nothing new as a computer interface. But the pieces were not in place for success until Apple's Siri ushered in an approach for using voice as a tool not only for searching but also for interacting with the online world in real time. APIs will be the channel in which voice finds the data and information it needs to fulfill requests made by mobile and web users around the globe.

Internet of things

While web APIs are transforming many industries, many opportunities are also emerging within the internet of things (IoT): internet connectivity applied to everyday objects. This new area of connected devices has an immense opportunity to impact our lives via automobiles, homes, and building connections, but it is also expanding to manufacturing with 3D printing and to health and fitness through the quantified-self movement. APIs will play a central role in connecting common objects to the internet, a territory that is largely unexplored.

Cloud trends

Throughout the commerce and social phases of API evolution, APIs were widely considered fun and games and not something businesses would pay much attention to as a serious approach to exchanging data and building apps. But with the introduction of cloud computing, web APIs became real: Developers could deploy global IT infrastructure using web APIs.

Maturity of IaaS layer

Amazon applied what it learned from success with commerce APIs and running its own internal infrastructure via APIs to delivering the essential compute resources developers needed. Amazon S3 for storage and Amazon EC2 for computer storage in 2006 were the seeds for the billion-dollar cloud computing industry of 2013.

In the early days only startups understood the potential of deploying Infrastructure as a Service (IaaS) using providers like Amazon Web Services. By 2013 the IaaS movement has matured beyond compute and storage to database, DNS, streaming, content delivery, and many of the other essential components that startups, the enterprise, and even governments need to be successful in this new technology-driven economy.

Opportunities in the PaaS layer

Infrastructure as a Service from Amazon has established a base for a new breed of technology solutions providers one level up: Platform as a Service (PaaS). This API-driven platform is providing a rapid deployment environment that developers are using to scale the next generation of web and mobile applications. PaaS providers are delivering an entire suite of essential services that developers need to focus on what they do best (i.e., the front-end features that users are demanding), leaving the backend operations to the experts. PaaS is the formula for whetting the huge appetite for web and mobile applications. PaaS, which is one of the fastest-growing aspects of web and mobile development, provides an API-driven suite of tools that provides essential services for developers in a pay-as-you-go approach. This is the next generation of business utilities designed for developer consumption.

Key takeaways

- Simplicity and intuitive resources are key aspects of modern APIs.
- Lightweight existing internet technologies are behind web APIs.
- APIs have been the driver behind social, cloud, and mobile technology.
- Devices, real time, automation, and voice are emerging trends in APIs.
- Easier cloud solutions for deploying web APIs are emerging.
- APIs and the cloud are the default for doing business in a digital economy.

About Kin Lane

Kin Lane has been working in the technology space for over 20 years. He has lived as a programmer, database administrator, architect, product developer, manager, and executive, with experience in business development, sales, and marketing, all while living with chronic entrepreneurialism. In this decade he has rolled all of that experience into a new persona, one that focuses exclusively on APIs, called the API Evangelist.

Lane considers himself a full-time evangelist not for a single API but for APIs in general. His focus is not just on developers but also helping the rest of the world understand the value and potential of APIs.

Follow his work on the business of APIs via apievangelist.com, the politics of APIs via apivoice.com, and his personal thoughts on kinlane.com.

About Gigaom Research

Gigaom Research gives you insider access to expert industry insights on emerging markets. Focused on delivering highly relevant and timely research to the people who need it most, our analysis, reports, and original research come from the most respected voices in the industry. Whether you're beginning to learn about a new market or are an industry insider, Gigaom Research addresses the need for relevant, illuminating insights into the industry's most dynamic markets.

Visit us at: research.gigaom.com.

© 2013 Giga Omni Media, Inc. All Rights Reserved.

This publication may be used only as expressly permitted by license from Gigaom Research and may not be accessed, used, copied, distributed, published, sold, publicly displayed, or otherwise exploited without the express prior written permission of Gigaom Research. For licensing information, please [contact us](#).