API

WHAT IS API

API: Application Programming Interfaces

An application programming interface (API) is a set of subroutine definitions, protocols, and tools for building application software.

A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer.

An API may be for a web-based system, operating system, database system, computer hardware, or software library. An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables, or remote calls. POSIX, Microsoft Windows API, the C++ Standard Template Library, and Java APIs are examples of different forms of APIs.

Just as a graphical user interface makes it easier for people to use programs, application programming interfaces make it easier for developers to use certain technologies in building applications. By abstracting the underlying implementation and only exposing objects or actions the developer needs, an API reduces the cognitive load on a programmer.

However, the notion of what API means has evolved significantly. APIs today are quite different from the application programming interfaces of old.

WHAT IS NOT API

- A piece of software: Software isn't an API (but it may render itself as an API to ease consumption of its capabilities).
- A user interface: A user interface isn't an API (but it may be built on top of one).
- A server: A server isn't an API (but it may host one or more APIs that expose the data and functions provided by the server).

IMPORTANCE OF API

API is a common language (lingua franca) between information systems.

API enables connectivity among systems. Similar to communication devices and systems enable connectivity among people. Thus, an enterprise without API can be considered as disconnected.

More important concerns include building omni-channel solutions, innovating faster than the competition, becoming a mobile enterprise, or operating in a hybrid cloud environment.

For many enterprises, public APIs are really the least of their business concerns.

To some degree, not having a public API today is like not having a website in the late 1990s.

APIs are a hot topic, energetically debated by business people, IT managers, and developers alike.

PRESENT SCENARIO WITH RESPECT TO API

APIs as business network enablers aren't new.

Banks have built payment infrastructures and clearing houses based on well-defined APIs for decades.

Modern APIs, however, are built explicitly for an open ecosystem (internal or external), not for closed private networks.

Whether your business was "born on the web" or has been around for 100 years, you're living in the age of cloud, analytics, mobile, and social computing where omni-channel has become table stakes.

To differentiate yourself from your competitors, you have to give customers an immediately engaging experience.

To deliver that experience, you need freedom to experiment and innovate.

CHARACTERISTICS OF GOOD API

Easy to use (easily consumable)

Well-defined interfaces

Controllable performance

Availability of performance analytics

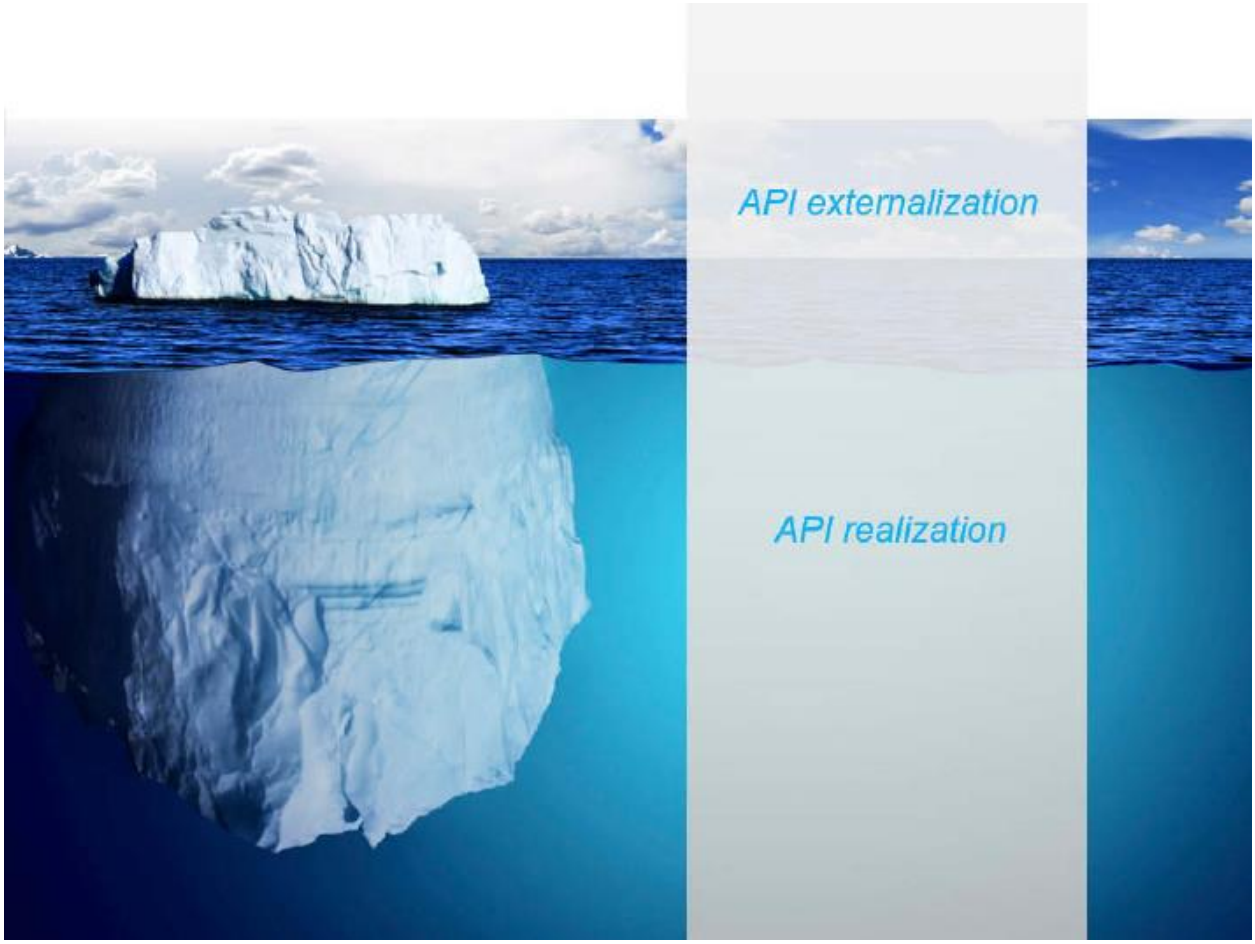
Optimized and updated regularly based on previous experience

Under the waterline are the business and IT concerns that make APIs practical to create, deploy, and operate.

These concerns include data mapping, security, rate throttling, monitoring, and version management.

A managed API not only has a well-defined interface and a defined target audience but also is under appropriately enforced business and IT controls.

Different groups have specific parts to play in API management



WHAT IS EASILY CONSUMABLE API

The consumption models for APIs are standardized with a focus on ease of consumption rather than ease of creation.

Developers want to use APIs for innovation and experimentation.

They're not as interested in how APIs were created (and at what cost) as they are in how easy the APIs are to consume.

Easy consumption doesn't refer only to what the API looks like.

To the developer who is API savvy, easy consumption also means that an API must be easy to find and easy to register to use, and it must be clear how much the API can be trusted in mission-critical solutions.

To an API consumer, a great developer portal is everything. To an API provider, managing the API externalization and sharing processes is only the tip of the iceberg (see Figure).

API IS LIKE RACING CAR

A race car is built from rapidly replaceable components with well-defined interfaces,

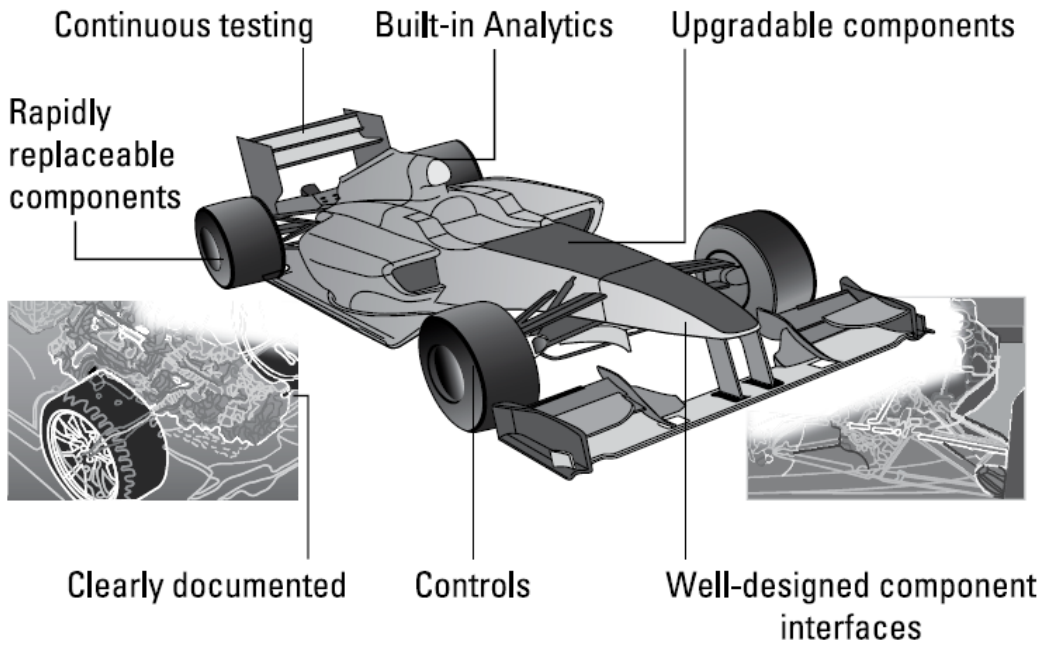
The car itself is instrumented with built-in controls and analytics.

Although parts of the car may remain stable throughout the season, some component is always optimized based on lessons learned in the previous race.

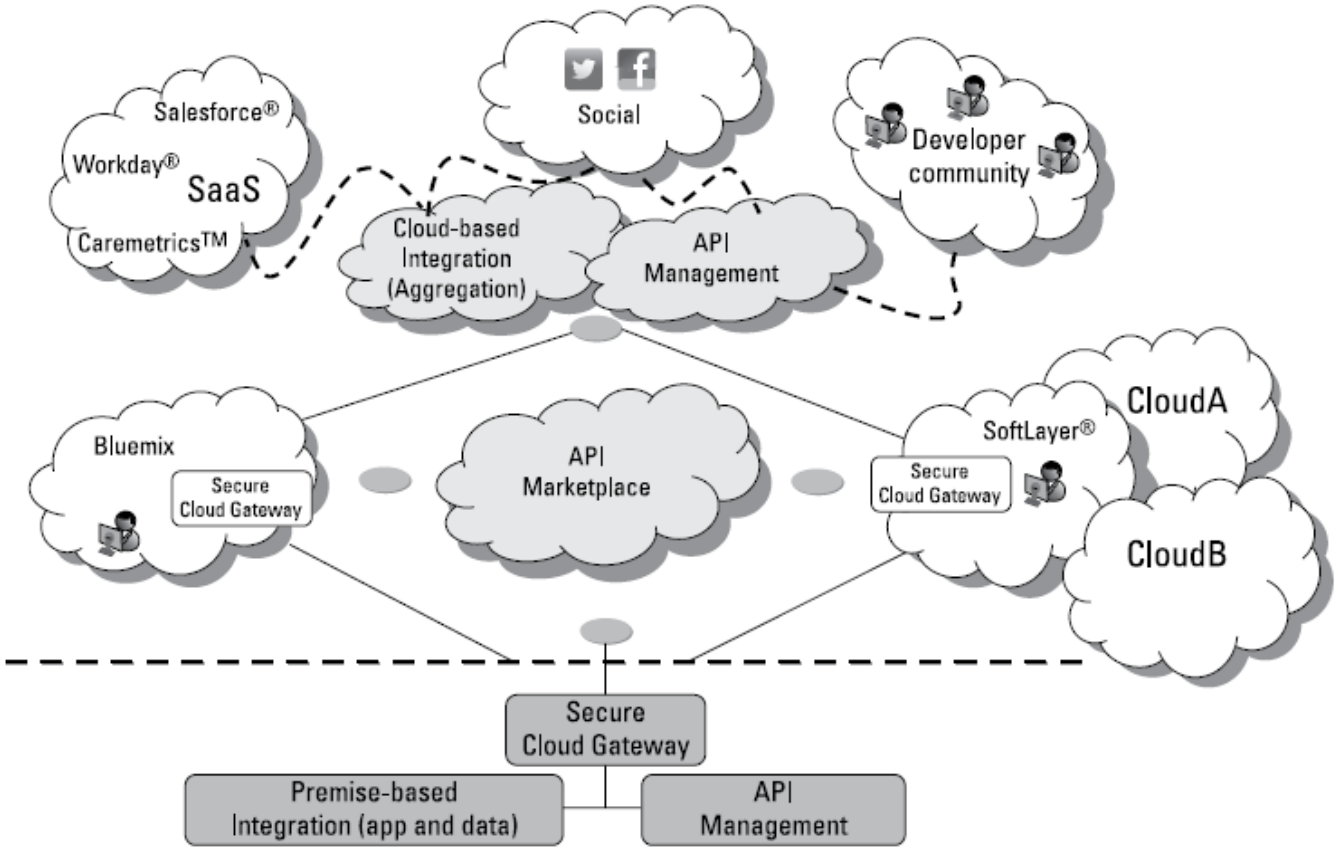
Modern enterprises are in many ways like Formula 1 teams, always trying to optimize the business model and always looking for the right balance between change and stability.

APIs are one way in which experimentation can be harnessed for enterprise advantage.

“Try early, learn fast, and scale easily” is a good principle to apply to the world of APIs.



HYBRID ECOSYSTEM WITH RESPECT TO API



HYBRID MULTI-CLOUD ENVIRONMENT

The enterprise trend toward deploying hybrid and multi-cloud environments is increasing and at the same time opening up abundant opportunities for cloud service providers (CSPs). With more than half of enterprises deploying hybrid cloud infrastructures, CSPs can step in to deliver either the private or public sides of the cloud equation. In some cases, large CSPs can handle both.

And since one cloud does not fit all, many enterprises are seeing the need for multi-cloud services to run their businesses. In fact, a recent IDC survey showed that within the next two years, enterprises expect to increase their average cloud service usage from two cloud providers to four.

It makes good business sense for CSPs seeking growth in the hybrid and multi-cloud markets to leverage the “cloud gravity” within global colocation data centers where there are dense ecosystems of private and public Infrastructure-as-a-Service, Platform-as-a-Service and/or Software-as-a-Service cloud providers. Many CSPs are leveraging interdependencies among private and public cloud offerings and interconnecting multi-cloud environments to add value to their services.

According to Gartner, hybrid IT is transforming IT architectures and the role of IT itself. Hybrid IT is the result of combining internal and external services, usually from a combination of internal and public clouds, in support of a business outcome. But meeting business and IT goals often require

more than just a hybrid or public cloud strategy. They require a comprehensive managed hybrid IT strategy.

Following terms imply similar meanings: “Multi-cloud based system”, “system of systems”, “cloud based hybrid ecosystem”.

API IN HYBRID ECOSYSTEM

Figure illustrates how a developer can use APIs to securely access any part of a hybrid environment.

In a “system of systems” world, there’s no traditional network perimeter to enforce, so interactions need to be controlled at the application level.

One of the most important aids for a developer is the catalog of APIs that are readily available for consumption.

Don’t show every single API out there (there are too many); show only the ones that are relevant to the developer in question.

That developer shouldn’t have to care about how and why the API is procured; she should focus solely on what she can do with the API after it has been made available to her.

In a hybrid world, the domain structure is inevitably complex.

Public marketplaces, private API catalogs, partner portals, and more are part of the API fabric.

Consequently, a well-defined community structure is more important than ever, with direct correlation between community design decisions and API design decisions.

At development time, the API marketplace provides information about the APIs that are available to the community of that developer.

At runtime, the cloud gateways secure the communication between the API consumer environment and any API endpoint, independently of location.

For internal API use, sharing is preferably managed on a community basis, making sure that given developers see only the APIs that their community is supposed to use.

Finally, API owners need business statistics about who uses their APIs and how much. These statistics measure success against the business objectives for the portfolio of APIs

Modern enterprises need API management as well as enterprise integration, but as two separate platforms aimed at different target audiences.

Do remember, though, that if you are already doing Service Oriented Architecture (SOA), you have a good set of assets that you can quickly discover, assemble into (proxy) APIs, and expose through an API management platform.

SOME GUIDANCE: IN THE CONTEXT OF LIVING IN A HYBRID WORLD

GOAL

The desired outcome is to empower both developers and IT operations. Careful controls need to be placed on the APIs made available, and any communication on an open network needs to be secure and appropriately managed.

The most-often-cited reason for not adopting a hybrid cloud approach is security concerns.

The second-most often-cited reason is fear of losing operational control.

AUDIENCE

The audience can be any mix of internal and external developers who are part of the hybrid ecosystem, so designing appropriate community structures is very important. Making decisions at the single-person level is highly impractical; you need a structure that allows you to make and enforce API sharing decisions at the community level, treating all developers in a particular community in the same fashion.

APIS TO PROVIDE

The APIs you need to engage the audience depend on the audience community structure.

For external audiences, you probably need a good set of predefined enterprise APIs.

For internal audiences, you need some opportunistic APIs to support rapid creation of innovative apps.

No hard-and-fast rule applies to all hybrid cases; each hybrid case tends to be different.

CONSUMING SOMEBODY ELSE'S APIS

Considering which third-party APIs to consume is harder, because over time, the number and variety of available APIs is dramatically greater.

The best advice is to start simple on API consumption.

Pick a small number of important APIs that you want to consume: social APIs, analytical APIs, mobile back-end APIs, or something else, depending on your most immediate business need. You also have the option of curating third-party APIs into your own simpler or more controlled version, so take that option into account in your decision-making.

From an IT operational perspective, complexity may become manageable due to the evolution in hybrid cloud platforms.

PROGRAM YOUR WORLD

The final API entry point focuses on the world of devices and machinery. It's based on two beliefs: The consumer-driven mobile revolution is about more than just phones, and manufacturing and logistics will drive the intelligent corporate Internet of Things.

Healthcare, utilities, cities, manufacturing — practically everywhere you turn, you see the need for a more intelligent approach to blending people, software, and machines.

The idea of programming everything into a single intelligent experience is still emerging, but signs of progress toward that goal are seen every day in such examples as smart cars, interactive retail experiences, and ecofriendly buildings.

One important difference separates this entry point from the others: You generally can't change a device after it exists in its physical form. That fact in turn means that the device APIs remain what they are.

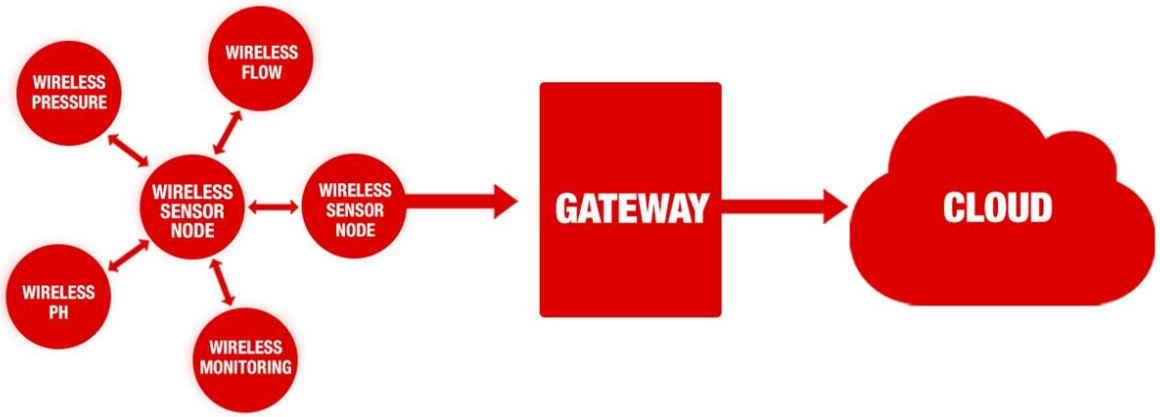
Therefore, the focus shifts to providing an enriched experience with optimized execution, as follows:

✓ Extend software into the physical realm

✓ Control any component, software, or device through programmable APIs.

✓ Use device-driven feedback and insight to optimize the behavior of the entire system, not just a single component.

✓ Where appropriate, monetize your high-level control capabilities as your own set of APIs



ROLES IN API MANAGEMENT

Different groups have specific parts to play in API management

BUSINESS OWNER

The API business owner decides the following:

- The plans (terms and conditions) under which the API can be consumed
- The communities that the API will be shared with
- Whether the API is succeeding in its objectives (if not, the business model needs adjustment)

IT OPERATIONS

IT operations must ensure certain operational characteristics, all of which can also be done without changing the API definition or implementation in any way. These characteristics are as follows:

- The runtime that hosts the API can be operated securely and robustly.
- The API is properly authenticated, and authorization is in place for anyone who uses it.
- API traffic is optimized and prioritized according to business needs.

API DESIGNER

The person that holds the API designer role physically creates and deploys the API. He needs to do the following:

- Define the API interface
- Discover back-end endpoints that may provide the data or function required to implement the API
- Configure the mapping between the API interface and the back-end data or function sources

API CONSUMERS

API consumers also need to make good decisions. In particular, they need to decide which APIs they're willing to use for what purposes and then ask the following questions about each API:

- What is the payment model for using the API and is that acceptable for your purpose?
- Will you need a corporate proxy in front of the API to handle licenses, payment, and the like, or will every developer register independently?
- Is the API secure and reliable for mission-critical purposes?
- Any historical records about how the API has behaved over time may add to consumer confidence in using it.

REFERENCES / EBOOKS

API for Dummies, IBM Limited Edition, 2015
http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WSM14025USEN

Integration Throughout and Beyond the Enterprise
http://www.redbooks.ibm.com/redbooks/pdfs/sg248188.pdf