**Basic API Management Will Grow Into Application Services Governance**

[Share this on Facebook](https://www.gartner.com/doc/2889617?ref=unauthreader)[in](javascript:void(0);)

**28 October 2014**

**G00271064**

**Analyst(s):** [**Paolo Malinverno**](http://www.gartner.com/analyst/11542)

**Free preview of Gartner research**

**Summary**

API management solutions have been maturing in the market for several years; however, the functionality that API requires is well beyond the basic set that recently introduced offerings include. IT leaders need to prioritize their API program requirements.

**Overview**

**Key Findings**

* API management, including its terminology, is becoming increasingly popular in application projects.
* APIs will be everywhere, and their use will rapidly increase in future applications.
* Beyond the obvious and significant technology differences, the concept of the API and service-oriented architecture are inextricably linked.

**Recommendations**

* Prioritize your needs for your API program in time, keeping in mind that successful API programs go well beyond initial API management functionality within a year or less.
* Beware of recent, API management offerings that are immature, functionally incomplete and self-defining.

**Table of Contents**

* [Analysis](https://www.gartner.com/doc/2889617?ref=unauthreader#a1230436674) 
  + [Myth: APIs Are New; Service-Oriented Architectures Are Old](https://www.gartner.com/doc/2889617?ref=unauthreader#a717784388)
  + [APIs Will Be Everywhere](https://www.gartner.com/doc/2889617?ref=unauthreader#a1928656012)
  + [The General Notion of API Management](https://www.gartner.com/doc/2889617?ref=unauthreader#a535411013)
  + [Why API Management Is Not Enough](https://www.gartner.com/doc/2889617?ref=unauthreader#a566180876)
  + [Application Services Governance Is the Market](https://www.gartner.com/doc/2889617?ref=unauthreader#a684388028)
* [Gartner Recommended Reading](https://www.gartner.com/doc/2889617?ref=unauthreader#a1340216415)

**Tables**

* Table 1. [Two Worlds (Seemingly) Apart](https://www.gartner.com/doc/2889617?ref=unauthreader#1043734804)

**Analysis**

**Myth: APIs Are New; Service-Oriented Architectures Are Old**

Many Gartner clients believe that service-oriented architecture (SOA) is old and APIs are new; however, this is questionable thinking, for several reasons:

* The concept of the API predates SOA.
* APIs and SOAs are very different: They are programming interfaces and application architectures, respectively.
* In many cases, what people really mean is "REST is new, and WS-\* is old," which is hard to disagree with; however, this is a technology, rather than an architectural, statement:
  + SOA is an architecture.
  + SOA can be implemented in various ways, including WS-\* or REST.
* There is a lot of SOA (old, new, implicit, explicit — name it any way you want) in the implementation of many REST APIs (e.g., microservices), and this won't be changing in the near future.

In the previous iteration of this research (see ["Govern Your Services and Manage Your APIs With Application Services Governance"](https://www.gartner.com/doc/code/245902?ref=ddisp) ), the API management and the SOA governance technology (although the phrase "governance technology" is frequently omitted) worlds were still separated by sharp differences. Table 1 below illustrates a few striking contrasts between the two worlds approximately two years ago.

| **Table 1.**   Two Worlds (Seemingly) Apart | |
| --- | --- |
| **SOA Governance** | **API Management** |
| SOA governance is mainly about services. | API management is mainly about APIs. |
| You must govern services throughout their life cycle to get value out of SOA. | You expose APIs to the Web, and you need to manage them to prevent chaos in their use and to make the new channel work. |
| SOA governance has unlimited faith in Java and Web services. | API management has unlimited faith in REST. |
| The more services you have, the better (wrong). | Generally exposes one or two APIs, then drives as much usage as possible before introducing new ones. |
| Technology is typically on-premises. | Technology is mainly delivered as cloud services. |
| Services are typically used within a company's firewall. | API is typically used across the Web by external entities and mobile programmers. |
| People feel the need for governance. | The entity exposing APIs typically needs management; the users of the API, especially mobile programmers, would steer clear of anything that has a governance tag. |
| SOA governance wants to open up, embrace, and eventually swallow API management. | API management views SOA as a "dinosaur" of the past and wants nothing to do with it. |
| SOA governance generally involves a "stick" approach (see Note 1). | API management generally governs with a "carrot" approach. |

Source: Gartner (October 2014)

Most of the contrasts above are not that sharp anymore, and have been toned down, frequently toward the API management terminology. For example, when clients talk about SOA services they tend to use the word "APIs." Web API providers frequently express the need for "governance." It will take another year, or possibly slightly more, for the terminology to become more consistent, such that the two worlds will become one.

It better be that way, because APIs have a big future.

**APIs Will Be Everywhere**

Strictly speaking, APIs have always been everywhere, from the times of assembler subroutines. The difference is that now a lot of APIs will be available to programmers well outside the group that designed them. With REST interfaces properly documented in developers' portals, most of the technology barriers to easy use have been removed.

The era of SOA started the idea that APIs will be everywhere. SOA services have been designed for years, in an effort to provide agility in application design and to lower application maintenance costs, as a consequence of service reuse. The domain in which services reside is generally one enterprise, where an illusion of control over application development processes gave hope (and sometimes substance) to companywide reuse and the associated benefits.

One of the taglines of Gartner Symposium keynotes two years ago was "every company is a technology company." APIs make this even more true. Starting from media and high tech, they have now moved into financial services, government and retail. During the past two years, a critical mass of companies and government institutions has been publishing APIs into their data and some functions of their IT systems to fuel business-to-consumer (B2C) innovation, enable mobile apps and take advantage of more-direct B2B interactions with their business partners. This will continue to fuel the so-called "API economy." However, that's not all; in fact, the best is still to come.

APIs are increasingly key to application strategies (see ["It's Time to Gradually Give Up Developing, Buying and Maintaining Applications"](https://www.gartner.com/doc/code/259283?ref=ddisp) ) and effective application governance (see ["Use the Demand for New Applications to Drive Application Rationalization"](https://www.gartner.com/doc/code/255823?ref=ddisp) ). However, most future API use will come from digital business applications and things, as in the Internet of Things (IoT; see ["Which New and Old Applications Will Enable Digital Business?"](https://www.gartner.com/doc/code/267140?ref=ddisp) ), and APIs will be firmly tied into business scenarios.

The more APIs that are out there in trading partners and specific enterprises, the greater the need will be to manage them, from the point of view of who publishes them and who uses them. Frequently, companies will do both. Also, innovating business processes and downstream systems will increasingly rely on APIs working as expected. Here is where API management comes in.

**The General Notion of API Management**

Gartner has gone into some detail regarding the definition of API management (see ["Run and Evolve a Great Web API With API Management Capabilities"](https://www.gartner.com/doc/code/247962?ref=ddisp&latest=true) ). However, the general public's idea of API management — one which is, admittedly, less precise — is becoming firmly established in today's application portfolios. In the view of most practitioners, API management is generally about publishing, promoting and managing the usage of APIs, frequently in the cloud or another scalable, on-premises environment, enforcing proper security when the data used or returned by the API is restricted or sensitive. It also includes API support documentation resources that are generally known as developers' portals.

That simple definition carries several implications. You need to make sure that the API is used by the developers first, and then by the applications, scenarios and personas it is designed for, and that it works properly. APIs are published to open new business opportunities (or consolidate existing ones), and they must be easily consumable in a secure, process-like fashion by the developers you are targeting. So API management features typically include:

* Operational management, security, format translation and the collection of metrics associated with the usage of the API (sometimes called "policy management")
* Discovery, documentation, developer access provisioning, testing and collaboration (in the so-called "developers' portal")

**Why API Management Is Not Enough**

Most organizations will be providing APIs and using other organizations' APIs as well. They will publish APIs to their internal users, to their business partners or to the public. They will also use APIs that technology providers offer them (e.g., to integrate with SaaS applications) or APIs that will advance their multienterprise business processes (e.g., to send an order for specific supplies). API management is useful to publish and track the APIs you offer, as well as to understand your dependencies on the APIs you use.

If you have tight deadlines to publish APIs to a specific developer's constituency or publicly, the API management basic set of functionality above is generally enough. However, how do you design and build the right API, before you publish it? (See ["A Guidance Framework for Designing a Great Web API."](https://www.gartner.com/doc/code/264400?ref=ddisp&latest=true) )

APIs give access to functionality and data. In some cases, a simple API can be implemented from scratch; however, more frequently, an API is built by putting together data that needs to be extracted from independently designed databases, or by aggregating functionality coming from different systems. This may already be in the form of a technology-specific API, with some business logic on top that makes the final API fit for use — for example, by a mobile app. Implementing an API is hard work, but even after you've done that, you're far from finished.

That's what SOAs have been trying to do for years, with different technologies and varying degrees of success. If your company is doing SOA, the services you defined are most likely building blocks for the APIs your company is likely to publish. The current notion of microservices and software-defined architecture (SDA; see ["Software-Defined Architecture for Applications in Digital Business"](https://www.gartner.com/doc/code/264171?ref=ddisp) and ["Use SDAS to Virtualize and Service-Enable Your Legacy and Packaged Applications"](https://www.gartner.com/doc/code/264205?ref=ddisp) ) are also fundamental patterns in API programs, especially in complex application infrastructures.

Like APIs, SOA services needed to be managed. This is where several companies have failed to reap the benefits of SOA. Services were not properly published, were difficult to change or reuse, and they didn't always go through technology or organizational barriers. SOA governance and the related technologies were always a difficult fix. However, there are many SOA governance lessons that API management can build on, and the two disciplines are inextricably linked (see ["Govern Your Services and Manage Your APIs With Application Services Governance"](https://www.gartner.com/doc/code/245902?ref=ddisp) ).

The degree of change that digital business applications and the IoT will require to keep addressing business moments is unprecedented (see ["Which New and Old Applications Will Enable Digital Business?"](https://www.gartner.com/doc/code/267140?ref=ddisp) ). APIs will have to support that, so API management platforms will have to collect a lot more usage data, help API providers to keep APIs relevant, and get value out of them. Good APIs are the ones that serve their consumers best.

APIs are assets: they need to be tracked through a life cycle, not just when they are in operation, and are being actively used. Support of full API life cycle management (planning, design, implementation, packaging/operation/versioning/maintenance and retirement) is essential when you provide APIs. When you use APIs, there are basic life cycle events you really need to know about — e.g., new versions or, more radically, retirement.

Because of the speed of change of application needs, new Mode 2 (see ["Bimodal IT: How to Be Digitally Agile Without Making a Mess"](https://www.gartner.com/doc/code/268866?ref=ddisp) ) dynamic governance policies will be required. In tomorrow's world, governance will have to be empirical, continuous and process-based, with strictly followed principles, continual references to the new reality (what has been learned so far, and feedback from the field), vital sense-and-respond capabilities, and awareness of Mode 1 contact points.

In addition to all of the above, there is additional API management functionality (sometimes delivered as managed services or consulting) that is useful in API programs:

* The planning and design of value-generating APIs
* The promotion of API usage, targeting all the communities of developers that might be interested in it
* The assessment of the real value (e.g., monetization) of the API, and the support of the various business models associated with it
* The instrumentation and analytics of the API usage

Basic API management will get your API program off the ground; however, in the long run, this won't be enough. This is why Gartner has introduced the concept of application services governance.

**Application Services Governance Is the Market**

Gartner defines a market as a set of end users looking for solutions to the same problem. It is our view that all the functionality introduced above is what companies will need for their API programs during the next few years. Hence, we introduced the concept of application services governance to unify basic API management, SOA governance, full life cycle management of APIs, new governance models and all the additional useful functionality around it. That is the market our Magic Quadrants will continue to rate.

**Gartner Recommended Reading**

*Some documents may not be available as part of your current Gartner subscription.*

["Govern Your Services and Manage Your APIs With Application Services Governance"](https://www.gartner.com/doc/code/245902?ref=ddisp)

["It's Time to Gradually Give Up Developing, Buying and Maintaining Applications"](https://www.gartner.com/doc/code/259283?ref=ddisp)

["Use the Demand for New Applications to Drive Application Rationalization"](https://www.gartner.com/doc/code/255823?ref=ddisp)

["Which New and Old Applications Will Enable Digital Business?"](https://www.gartner.com/doc/code/267140?ref=ddisp)

["Run and Evolve a Great Web API With API Management Capabilities"](https://www.gartner.com/doc/code/247962?ref=ddisp&latest=true)

["A Guidance Framework for Designing a Great Web API"](https://www.gartner.com/doc/code/264400?ref=ddisp&latest=true)

["Software-Defined Architecture for Applications in Digital Business"](https://www.gartner.com/doc/code/264171?ref=ddisp)

["Use SDAS to Virtualize and Service-Enable Your Legacy and Packaged Applications"](https://www.gartner.com/doc/code/264205?ref=ddisp)

["Bimodal IT: How to Be Digitally Agile Without Making a Mess"](https://www.gartner.com/doc/code/268866?ref=ddisp)