SOLUTION BRIEF
CA API MANAGEMENT

Deploying Your API Infrastructure: Cloud, On-Premises or Both?

The quick answer is: It depends. There are **many** reasons why an enterprise might require an on-premises deployment of its API management platform. Likewise, there are many reasons why a cloud deployment makes sense. And sometimes there's a gray area where some combination of both may make sense. CA API Management provides multiple deployment options, depending on the needs of the enterprise. In this brief, we'll look at the pros and cons of each, from various points of view.

Executive Summary

Challenge

Deploying an API Management platform may not be a simple choice of “we’ll install it here” (wherever “here” happens to be). For many enterprises, this can become a complex question with equally complex answers.

Opportunity

Typically, when enterprise architects think of API management, they have a good definition in place of what that is and what it includes. While the components of API management are clearly important, deployment options can be equally important. And that’s where things can get confusing.

An API management platform typically provides:

- An API gateway that integrates everything while delivering end-to-end security for apps, mobile, the Internet of Things (IoT), and the cloud.
- A developer portal that allows enterprises to securely expose and manage APIs to internal or external developers. Depending on the vendor, this portal may also help developers discover what APIs are available, learn about them through documentation, interact with them through test calls and subscribe to them. This also may be a centralized hub for business users to manage accounts and for operations to monitor utilization.
- An administrative console that supports configuration of the system, monitors both the gateway and the portal—and, depending on the vendor—maintains API lifecycle management.

As noted above, that’s commonly the way people look at API management. But, I submit there’s another way—viewing API management from a usage model perspective. This can be broken into two phases:

- Design time—Making the API available for consumption for developers: to discover, learn about, interact with and subscribe to APIs.
- Runtime—Actual consumption of the API and resources, and enforcement of runtime policies on inbound and outbound transactions.

Let’s take a look at these two in a bit more detail.

Section 1:

Usage Models

Design time

At design time, an API provider's goal is to build and maintain a business relationship with the app developer, whether that developer is internal or external. They are offering a service with value to an app developer, providing the tools necessary to consume that API (documentation, tutorials, sample code, example apps), and promising to maintain certain levels of quality and operational stability. The developer is looking to discover what APIs are available, learn about them through documentation and interactive testing and subscribe to a plan that allows the developer to use the APIs in their application.

So, for design-time infrastructure, should you deploy on-premises or in the cloud? It depends.

Many enterprises want to support:

- **Secure zone initiatives.** If the business is undertaking an internal development project that they don't want exposed via a public interface, an on-premises portal is their only option.
- **Custom portals.** If the business requires extensive customizations built into the portal (e.g., built on top of the existing CMS system), an on-premises may be the best choice.
- **Multiple business divisions.** For customers that have one brand but multiple business divisions and would like a single portal that allows each division to publish and manage its own APIs, an on-premises solution is a good choice.

But for many other enterprises, deployment in a cloud model just makes sense—who wants to stand up infrastructure that they don't have to? Additional cloud-based portal benefits include reduced capex with opex spread out based on consumption and faster time to market (see Capex vs. Opex callout).

Design time operations are also not generally mission-critical, so if there are short downtimes due to a hiccup at a hosting provider, or an occasional failed browser request, there's no real loss. These are human interactions, so you refresh your browser page and move on.

Capex vs. Opex

Capex—capital expenditures necessary to implement a solution. This typically includes hardware to be deployed, software and server rack space. Typically, these are capitalized up front and depreciated over a number of years.

Opex—operating expenditures necessary to maintain uptime for a solution. In other words, ongoing expenses, such as leases (whether hardware or software), wages, supplies, maintenance/repair and utilities. Typically, these are expenses that are deductible in the year that they occurred.

Section 2:

Runtime

One way to get a mental image of API management is to think of plumbing. The portal is the pretty interface—your fancy sink, mirror, shower wall, etc. The gateway—the runtime engine, the plumbing—is really where the grunt work takes place. In your runtime environment, these are just some of the processes taking place in real time: API data and services handling, transformation/composition, routing/traffic control, security (including integration with an existing identity and access management system, SSO, OAuth/OpenID Connect implementation, policy-based authentication, and DLP, attack prevention), API lifecycle management, health tracking, performance and workflow, analytics, and reporting.

So is the best place to deploy on-premises or in the cloud? It depends.

For many customers, runtime is where their demons exist—such as sensitive data being exposed and fear of losing control of data in transit. They demand that anything coming in that touches important data/systems at runtime—mission-critical, secure systems—be on-premises. Enterprises want the "warm and fuzzies" of the runtime engine doing all this work directly under their thumb—which makes sense (see APIM Platform On-Premises vs. Cloud callout).

But for many other companies, runtime in the cloud is not a concern. Again, who wants to stand up an infrastructure and maintain and upgrade it if you don't have to? And yes, that capex issue is a big one for lots of organizations.

APIM Platform On-Premises vs. Cloud

There are several advantages of an on-premises deployment:

- Full control of data
- Full control of technology
- Complete ownership
- Upgrades on your own schedule
- Uptime driven by the enterprise

Likewise, there are several advantages of cloud-based deployment:

- Flexible geographic deployment
- Cloud provider infrastructure features
- No ownership (no capex)
- No upgrade worries
- Clearly defined SLAs

Section 3:

Design Time and Runtime

Confused? Ok. With design time and runtime in mind, let's look at the original API management view:

Design time (API portal)

As stated earlier, this is a business relationship between the API owner describing their service (technical contract) and available API plans (business contract), and the consuming developer subscribing to a plan (accepting the business terms) and building their application to provide the expected request and credentials (accepting technical spec).

Runtime (API gateway)

As noted above, this is where the action happens. This a technical exchange of business data to and from the consuming application (now in an end user's hands) and the API backend deployed on the enterprise's infrastructure (as built by a developer or configured by a gateway policy author).

With that in mind, let's take a look at design time and runtime side by side for one last perspective.

Design Time	Runtime
Very commonly public (Open APIs) and meant to be consumed by humans	Very commonly private (protected in the DMZ) and meant to be consumed by applications
Latency has very little bearing	Latency is very important—if data is on-premises, routing transactions through the cloud adds latency
Low volume (clicks)	High volume (transactions)
Developer community management	Policy enforcement
Browser (human) interaction	API (machine) interaction
Uptime important but usually not mission-critical	Mission-critical interface requiring high availability and disaster recovery

As you can see, there are some interesting comparisons when viewed side by side. Based on your environment, these comparisons are a good start to help you determine your deployment option.

The last point is very important to note, though, as you consider your deployment options.

Deploying your gateway in the cloud may result in latency issues, as noted in the chart—this is very dependent on your environment. Regardless, your gateway is almost always a mission-critical component of your infrastructure. Design time (the portal) is usually important, but if there's a network hiccup, you can usually just refresh your browser and move on. Not so much with the gateway.

CA Rankings by Industry Analyst

As evidenced in the following reports, only CA Technologies is rated as a leader by all four industry analysts that monitor the API management sector.

- Gartner Magic Quadrant[1]
- The Forrester Wave[2]
- KuppingerCole Leadership Compass[3]
- Ovum Decision Matrix[4]

Section 4:

The CA Technologies Advantage

When you're looking at deployment for your API management solution, CA Technologies has you covered—providing a platform for you that meets your needs. Options include:

CA API Gateway/CA Mobile API Gateway/CA API Management SaaS. Available on-premises or via the cloud, CA API Gateway and CA Mobile API Gateway are market-leading solutions in API Management, and the only gateway solutions ranked as a leader by all four major industry analysts (see CA Rankings by Industry Analyst callout).

CA API Developer Portal. This solution is the hub for:

- API owners to expose and publish APIs
- Developers to find APIs, learn how to use them and generate code to insert in their application for API utilization
- Business managers to manage API access for enterprises/developers
- Operations to conduct analytics on API utilization

This hub can be installed on-premises, in the cloud or in a hybrid mode where the portal is deployed in the cloud to manage on-premises gateways—bringing true freedom of choice to the enterprises grappling with the question "Where do I deploy my API Management solution?"

To learn more about CA API Management, visit **ca.com/api**

ca technologies

Connect with CA Technologies at ca.com

CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate—across mobile, private and public cloud, distributed and mainframe environments. Learn more at **ca.com**.

[1] Paolo Malinverno, "Gartner Magic Quadrant for Application Services Governance," April 9, 2015

[2] Randy Heffner with Christopher Mines, Eric Wheeler, "The Forrester Wave: API Management Solutions, Q3 2014," September 29, 2014

[3] Alexei Balaganski, KuppingerCole Report, "Leadership Compass," July 2015

[4] Saurabh Sharma, "Ovum Decision Matrix: Selecting an API Management Solution, 2016–2017," April 11, 2016