WHITE PAPER | SEPTEMBER 2014

Choosing the Right API Management Solution for the Enterprise User

The API Opportunity

The application programming interface (API) may be an old concept but it is one that is undergoing a transformation as, driven by mobile and cloud requirements, more and more organizations are opening their information assets to external developers.

Today, 75% of Twitter traffic and 65% of Salesforce.com traffic comes through APIs. But APIs are not just for the social Web. According to ProgrammableWeb.com, the number of open APIs being offered publicly over the Internet now exceeds 12000—up from just 32 in 2005. Opening APIs up to outside developers enables many technology start-ups to become platforms, by fostering developer communities tied to their core data or application resources. This translates into new reach (think Twitter's rapid growth), revenue (think Salesforce.com's AppExchange) or end user retention (think Facebook).

The use of APIs for sharing information and functionality with outside developers is not limited to technology start-ups. More and more enterprises, driven by cloud, mobile and partner integration initiatives are using APIs to put themselves at the center of a developer ecosystem and—in so doing—driving new reach, revenue and retention possibilities around their information assets. However, unlike many start-ups, enterprises must approach API publishing with great caution. They have a good deal on the line, including reputation, regulation and the simultaneous needs of customers, partners, employees and shareholders.

The Enterprise API Management Challenge

Publishing APIs to an external developer community, be it partner or public, introduces a number of challenges and risks for the enterprise. How do you protect the information assets you are exposing from abuse or attack? How do you deliver your APIs as reliable services with no downtime that can impact your API users? How do you govern access and usage of your APIs in a consistent, policy driven way? How do you make money from your APIs? How do you help developers discover your APIs and self-manage their access? While these questions are relevant to start-up and enterprise alike, they are more acute and urgent for enterprise IT organizations. Enterprises cannot afford the reputation damage that may result from a rushed API Management strategy. They have deliberate IT processes and safeguards that need to be upheld.

But no matter what type of API an enterprise wants to expose, it will need an API Management solution that can address some basic functional areas:

- **API Security**—Enterprises cannot afford misuse or abuse of their information or of any application resources exposed by an API.
- **API Lifecycle Management**—Enterprises need a way to ensure API updates do not break when they upgrade/version APIs or move between environments, geographies, datacenters and the cloud.
- **API Governance**—Enterprises need a way to control and track the broader operational character of how APIs get exposed to different partners and developers, through policy characteristics like metering, SLA, availability and performance.
- **Developer Enablement and Community Building**—Enterprises need a way to bring developers on board, manage these developers and assist them in making the most of the exposed APIs.
- **API Monetization**—For some enterprises, publishing APIs is not enough. APIs also represent a new revenue opportunity. Different API Management solutions enable monetization to different degrees.

For enterprises, addressing these functional requirements is non-negotiable. However, along with these functional requirements, an enterprise will expect its API Management solution to deliver certain operational characteristics relevant to its unique IT experience.

- **Solution Security**—Since API Management solutions get deployed in the DMZ, enterprises will also need robust IT-class API solutions that can meet a range of security requirements, from penetration protection to PCI compliance to FIPS to HSM support for API key security.
- **Solution Manageability**—Enterprises have development, test and production environments that span geographies, datacenters and clouds. They will therefore need an API Management solution that can fit their specific development styles and processes.
- **Solution Reliability**—Enterprises publishing APIs commercially expect 5 9's uptime, if not greater. Enterprises cannot afford outages. What are the characteristics of a robust and available solution?

This white paper examines these different functional and operational requirements, to give IT managers, Web managers and enterprise architects key information for selecting an API Management solution.

API Management Solution Functional Requirements

API Security

When looking for an API Management solution, security features are often top of mind for prospective buyers—not least when the buyer is an enterprise looking to protect vital information exposed through an API independent of standards like SOAP, REST or JSON. API security concerns begin with access control. For externally facing APIs, this means having the ability to:

- Accept different kinds of credentials for authentication
- Issue different kinds of credentials to developers
- Support different "resource" authorization schemes including federated ones like OAuth and SAML

For enterprises this challenge is compounded by the need to integrate with existing identity infrastructure. Therefore, the overarching goal is to achieve both flexibility and integration. In policy there should be an ability to support different kinds of access tokens and even move from one kind of developer API key to another, without touching code. The solution should be able to support a wide range of OAuth schemes (given the standard's growing importance for APIs) but also handle a variety of OAuth styles like HMAC and combinations with enterprise standards like SAML. Of course, the API Management solution also needs to work with pre-existing identity investments from companies like Oracle, IBM, CA and RSA.

However, API security doesn't stop at access control. APIs provide the programmatic window into your data. For that reason, an enterprise-class API Management solution will need to give the enterprise architect or security administrator fine-grained control over what data get exposed, how this information is kept confidential and how its transmission can be guaranteed against interception or tampering.

Lastly, API security rests on the integrity of both the API and the data/functionality it exposes. This requires an ability to ensure APIs are not compromised by attack, denial of service or misuse. A good API Management solution will equip its operator with a wealth of threat protection controls that will assure the availability and fidelity of the API and the communications it enables.

API Lifecycle Management

APIs are not built in a vacuum. Like any application functionality, APIs demand their own development lifecycle, from design to coding to testing to deployment. This requires an ability to track changes to an API across the development lifecycle, whether the development process follows a waterfall or agile approach. For this reason, any API Management solution aiming to meet the needs of an enterprise will need to have fully-functional workflows for:

- Promoting APIs from development to production
- Managing the associated policy metadata
- Restricting change contributions using an integrated RBAC control
- Accommodating approvals and rollbacks

A fully-functional API Management solution should also be able to accommodate multiple versions in production simultaneously, either to accommodate older clients or to accommodate different access technologies like SOAP, REST and JSON.

However, a lifecycle management framework that can only accommodate localized development will not meet the needs of most modern enterprises. With the growing importance of the cloud, both public and private, enterprises will require an API Management solution that can span testing and production in the cloud. This will require an ability to isolate API developers from the vagaries of network idiosyncrasies and topology.

API Governance

Governance is a broad term often used to capture a wide range of management, process and visibility requirements. It defines the terms and conditions under which an API is exposed to one or more consumers. While "governance" encompasses security and lifecycle concepts, it also articulates various SLA, monitoring and reporting requirements. Furthermore, in the case of API Management solutions, it is relevant to the broader imperative of enabling differentiated terms and conditions for sharing API data and functionality to different consumers based on their identity, capability, subscription level or other transactional context that can be defined in policy.

Effective API governance is all about flexibility. The technology for controlling how APIs get shared should follow the preferences and processes of the enterprise and not the other way around. This means that an API Management solution should be configurable around any SLA, security, log or other control using policy. Policy is at the heart of flexibility and assures consistency from one implementation to the next. API Management solutions that constrain administrators to course-grained controls without a full policy IDE limit what can be governed and how it can be controlled.

Developer Enablement and Community Building

Governing an API ensures consistent control for the publisher. However, if that API cannot be easily discovered and consumed by external developers, the publisher risks that it will go unused. For that reason, most modern API Management solutions go beyond control features like security, lifecycle and governance to provide functionality that helps publishers expose information about their APIs to outside developers—often via developer portals. A developer portal provides a single point of interaction for the developer to register for an account, request an API access key, discover what APIs are available and see example code.

An API developer portal focused on enterprise usage should:

- Support different classes of external developers (e.g. the publisher should be able to attribute different rights to partner developers and public developers).
- Provide various self-service capabilities (e.g. subscription levels and rate plans).
- Give developers visibility into their API usage and key performance metrics (e.g. response time).
- Allow developers to share best practices through community features (e.g. a forum).

Since different enterprises will come to API publishing with different experiences and priorities, a one-size-fits-all API portal approach will be no more attractive than a one-size-fits-all API security, lifecycle and governance framework. For this reason, many enterprises will want to consider a decomposable API portal. This could mean a white-label portal that can be customized to suit a particular developer engagement strategy. It could also mean an API portal that can be consumed as discrete components by a pre-existing enterprise developer portal. Again, flexibility is the watchword.

API Monetization

Related to the idea of developer enablement is the concept of monetization. While many enterprises will want to foster adoption by allowing free access to their Web and mobile APIs, others will want to offer pay-per-use options for higher tiers of access. Again, there is no single right way of approaching the monetization problem. Some options are:

- A "freemium" model where usage below a certain threshold of data transmission or client requests is free
- Charging for specific levels of service guarantee or for priority over free users
- Offering premium information or functionality unavailable to non-paying customers

Regardless of which approach is taken, the API Management solution should be sophisticated enough to give an enterprise flexibility in how it sets up its revenue criteria. The solution should be able to:

- Capture a range of usage statistics, to create a basis for measuring consumption
- Provide advanced SLA and Class of Service capabilities, allowing for traffic prioritization
- Compose virtual pay-only APIs that could be isolated for paying customers, without coding

API Management Solution Operational Requirements

Solution Security

Since an API Management solution will often be the only piece of technology separating enterprise APIs from the outside world, the level of security the solution can confer on APIs will only be as strong as the security of the solution itself. If the solution is compromised, any security rendered onto the APIs will be similarly compromised. Therefore, enterprises examining API Management solutions should make the solution's security an absolutely critical consideration.

Since these solutions will be interposed as intermediaries between the outside world and internal APIs, the first quality often evaluated is whether the solution itself can be compromised. This will depend on what kind of penetration testing the solution has undergone, how constrained access to the solution is and whether it has met key vulnerability assessments. Consideration should be given to STIG tested solutions, PCI DSS certification for solutions that will pass credit card information, FIPS compliance and Common Criteria certification for solutions that need to meet higher government security standards.

For most practical purposes, enterprises will often look at API Management solutions that are proxy based for handling the intermediation of outside requests to an internal API. Intermediary- based API Proxies offer the advantage of clear inline points of control and isolation, making security certification and administration simpler (just like with network firewalls). Some may also offer onboard HSM support for encrypting API keys. Since API keys in many scenarios are the main line of authentication defense against abuse, protecting those keys from theft through encryption is a prudent strategy.

Solution Manageability

Unlike a typical startup, which may run its entire production Web site from a single Amazon instance or small hosted provider, an enterprise will typically have varied development and production environments. For example, an enterprise may have:

- Geographically-distributed developer teams
- Production environments that span global datacenters
- Cloud-based disaster recovery systems

Therefore, manageability will be central to any selection decision. Considerations like how you manage clusters of API proxies, how you load balance geographically, how you operate in a lights-out datacenter environment and how you handle peak loads will take priority over other features. Again, not all API Management solutions are designed to cater to the specific needs of the enterprise, so care should be taken in evaluating how various solutions support cluster management, fail-over, load bursting, disaster recovery and other operational management factors before embarking on a particular path.

Solution Reliability

Once an enterprise decides to embark on an API publishing program, it will effectively become a service provider to its API consumers. These consumers will come to rely on the enterprise and expect continuous uptime. In this context, an enterprise will inevitably place a considerable premium on reliability when selecting its API Management solution. The enterprise will look for solutions where redundancy is built in and risk of downtime has been extremely minimized, if not eliminated. The need for continuous uptime may eliminate hosted or cloud-based options. While not inherently unreliable, most pure-play cloud API Management solutions are run by small companies that tend not to provide the kind of mission-critical redundancy and support enterprises have come to expect. For that reason, enterprises looking at API Management solutions may want to consider only those solutions that can:

- Be deployed in the enterprise's own datacenters and private cloud
- Meet the kind of high availability, redundant configuration that would guarantee continuous uptime

Conclusions

No two enterprises have exactly the same needs or environment. Therefore, there will never be a one-size-fits-all API Management solution. However, all enterprises share a common need for excellence in functional capability and operation. For most organizations endeavoring to start publishing APIs externally, this will translate into a desire for a flexible, policy-driven API Management solution that can meet the production rigor of a dial-tone class service provider. Functionally, it will require an API Management solution that can meet a variety of security pre-requisites, accommodate common development lifecycles, be governable through policy, enable developer onboarding, foster developer engagement and support the option of monetization. Operationally, the API Management solution should be secure, manageable and reliable.

Contact CA Technologies

CA Technologies welcomes your questions, comments and general feedback.

For more information please contact your CA Technologies representative or visit **www.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**.

 CS200-87227_0914