DZone
A DEVADA MEDIA PROPERTY

BROUGHT TO YOU IN PARTNERSHIP WITH WSO2

Open Source API Management

CONTENTS

WRITTEN BY NUWAN DIAS
SENIOR DIRECTOR, WSO2

In this Refcard, we discuss the open source advantage and how it helps to deliver a modern-day API management solution. We look at the strategic value of APIs to enterprises and examine how open source becomes relevant to these strategies. We also look at the technical benefits of open source in terms of deploying future-proof API management architecture.

APIs and Their Relevance to Modern Enterprises

Modern enterprises increasingly expose their services as APIs for better reusability and faster innovation. This causes APIs to live at the heart of many systems and has led to widespread API adoption. APIs enable organizations to compose their capabilities as reusable units. These units can then be delivered independently and help organizations to rapidly increase their innovation cycles. This is somewhat similar to how automobiles are developed by using a large number of parts that are independently manufactured. These parts are then brought together to form different car models. For example, we see the same engine type being used in different models from the same manufacturer. This has helped the automotive marques, and the industry as a whole, to innovate rapidly. Car manufacturers are adroit at coming up with newer models while efficiently maintaining the older ones. This is made possible by independent and reusable units.

The Strategic Role of APIs for Sustained Growth

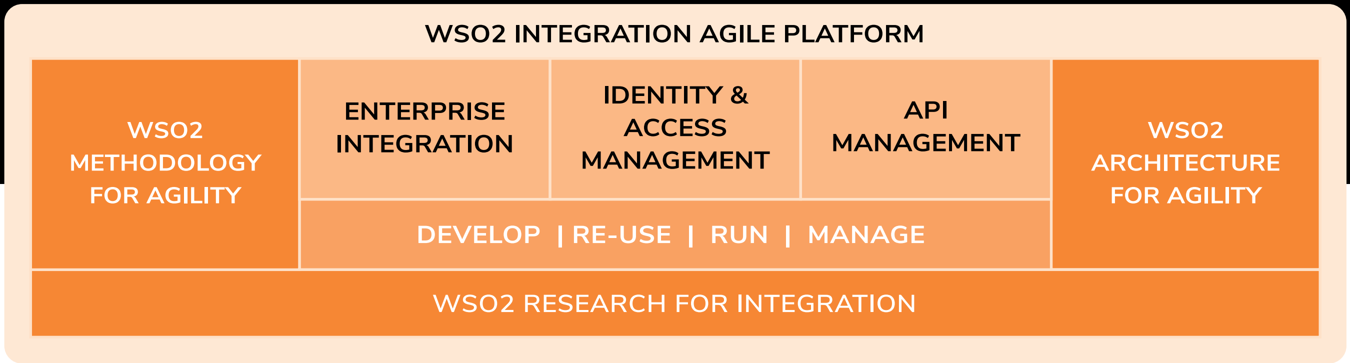
Most readers should be familiar with the popular Bezos Mandate. This occurred in 2002 when Jeff Bezos, Amazon's CEO, issued a mandate that required all teams in Amazon to expose their data and functionality through interfaces (APIs). Quoting the first item from his mandate, he said, "All teams will henceforth expose their data and functionality through service interface."

This mandate is most likely responsible for Amazon's highly successful journey so far. The company has built amazing technologies thanks to all their functionality being exposed through APIs, allowing creative individuals to take advantage of what's already been built. In addition to helping engineers build great technology, these APIs have also given Amazon the flexibility to innovate by adapting to market demands. The ability to adapt and innovate is key to an organization's sustained success and growth.

WSO2

Open Source Integration and API Management for Digitally Driven Organizations

All digital transformation is now API-driven, and integration technologies underpin their evolution. You can deliver faster, lower-risk integration projects with WSO2's open source Integration Agile Platform — including API Management, Enterprise Integration, ESB, and Identity Management technologies. To help make your integration an agile process, WSO2 provides unique technologies, methodologies, and architectures to speed up your transformation.



WSO2 INTEGRATION AGILE PLATFORM

WSO2 METHODOLOGY FOR AGILITY

ENTERPRISE INTEGRATION

IDENTITY & ACCESS MANAGEMENT

API MANAGEMENT

WSO2 ARCHITECTURE FOR AGILITY

DEVELOP | RE-USE | RUN | MANAGE

WSO2 RESEARCH FOR INTEGRATION

The WSO2 Difference

First-in-the-industry integration agile methodology

Our agile transformation methodology helps IT transform to an integration agile model. You get faster releases and quicker responsiveness to the business.

Unique open source technology and licensing

Our approach to open source ensures the benefits of faster innovation, community contributions, freedom from cloud lock-in, and the value-add of support patch binaries.

Broadest integrated platform

No need to mix and match different API management, integration, identity, analytics, or microservices technologies from multiple vendors. We offer a common architecture across all.

FORRESTER®

WSO2 Named a Leader in The Forrester Wave™: API Management Solutions, Q4 2018 Report

wso2.com

Open-source technologies have key characteristics that make them strategically better for development than to closed-source, proprietary solutions. Let's take a look at some of these benefits.

LIMITLESS INNOVATION

One of the key advantages of open-source technology is the limitless creativity that drives limitless innovation. As an organization that operates in the modern world, you can never really predict the future. The best bet, therefore, is to reasonably predict what the future is going to be and equip yourself to be able to adapt fast, when necessary. This is one key reason why open-source technology is a great choice. It gives an organization the flexibility to change fast.

APIs — which are at the forefront of your digital assets — need to be flexible to adapt fast to cater to new market demands. This sometimes requires new capabilities to be added to an API management solution. An open-source API management solution gives an organization this flexibility.

A closed-source, proprietary API management vendor would have limited resources. However, owing to the strength of the community, an open source vendor has limitless resources. This also gives a user the ability to contribute by adding new capabilities. This way, the user does not solely rely on the vendor offering the solution but has much more freedom and flexibility to implement the required capabilities.

TRANSPARENCY

Any profit-driven organization must stay ahead of its competition; this requires creativity and forward-thinking. The organization's technical leadership needs to have a strong vision for its future. They need to be creatively thinking about newer technologies to beat their competition, trying out newer technologies and be courageous enough to evaluate new technology and make decisions on their own. This requires new technologies, even if in their pre-release status, to be available for evaluation and experimentation.

APIs, which are a key value addition, play a major role in shaping an organization's future. It is therefore crucial for a user of an API management system to understand more about how it works, the thought processes and developments happening around it, and plans for the future. This requires the API management system to provide full transparency into its code, discussions, and roadmaps. The benefit of an open-source API management system is that all of its code, discussions, and roadmaps (in certain cases) are open to the public. This is a significant win for its users. It gives them the opportunity to plan well ahead of time based on what is to come and align their strategies to stay ahead of the curve. It also gives them the opportunity to influence the roadmap. The roadmap being public and open for discussion gives its users the opportunity to influence its progression.

QUICK TO GET STARTED

One of the key requirements in any API project is the ability to quickly get your hands dirty by using the API technology. Being able to evaluate API management systems based on first-hand experience instead of RFPs, presentations, and videos is extremely valuable. Trying out and creating your first few APIs and consuming them gives you the confidence in what exactly the API management solution has to offer, instead of relying on hopes and promises. Being free from licensing-related issues and restrictions and not having to go through sales calls for an initial proof of concept are significant wins in terms of being able to start quickly.

THE ABILITY TO START SMALL AND SCALE

Most API programs start at a small scale and eventually grow into massive businesses. Open source business models are much friendlier when it comes to starting small and growing compared to closed-source, proprietary business models. Most open-source software providers earn revenue via commercial support. This is great for community users of the software, as it gives them the flexibility to use the software at no cost to begin with, given they rely on community support only at first. As their API programs succeed and become critical parts of their business, they have the option to obtain paid support from the vendor with SLAs. These types of business models are heavily conducive to making API programs succeed compared with traditional enterprise licensing-based models.

SECURITY

The popularity and success of APIs have made them a rich hunting ground for attackers. As the number of APIs being exposed by organizations grows exponentially, their threat landscape grows as well. Luckily for them, the transparency provided by open-source API management systems guarantees a certain level of trust in the security offered by these products. The fact that the code for these systems is open makes it easier for security scan tools to perform code scans to check for known vulnerabilities and anti-patterns. Anyone willing to perform these scans can simply check the code and run scans themselves, instead of merely relying on the vendor's word.

Almost all software systems heavily make use of third-party libraries. There are many such libraries that contain known security vulnerabilities. When a software system uses open-source code, it makes it possible to discover the libraries that the software is using and check them for known vulnerabilities.

The simple fact that there is a community of developers with different types of expertise and backgrounds working on open-source software systems, itself, makes the code much more secure and better than a closed-source solution.

"Given enough eyeballs, all bugs are shallow," said Eric Steven Raymond in his essay, "The Cathedral and the Bazaar." He dubbed it "Linus's Law," in honor of the creator of the Linux operating system, Linus Torvalds. The message being delivered here is that, given a large enough developer/tester base, almost every problem in a piece of software will be characterized quickly and the fix will become obvious to someone. This is the simple benefit of having better software reviewing processes. The review audience of open-source software systems is much larger and diverse. This not only makes the software better and more secure, but, in case of a defect or vulnerability, it also enables finding a fix faster.

The Open Source Advantage in API Management Capabilities

As explained in the previous section, APIs are strategic to an organization. Therefore, the management system in place to govern these APIs becomes heavily strategic as well. You need to make sure you leave enough room for expansion to cater to current and future needs when choosing an API management platform. The platform you choose should also be flexible in catering to your organization's specific needs. Unlike commercial off-the-shelf (COTS) solutions, open-source software is generally good at various types of customizations that make the solution a better fit to cater to your requirements.

An API management system helps organizations build and monetize their digital assets. This requires the API management system to be connected to various other systems within the organization. Compared to closed-source enterprise software, open-source software is generally built on open standards and technologies, reusing a lot of commonly used libraries. This makes open-source software much easier to integrate with other software systems. Contributions from the open source community also make it easier to build newer integrations. Let's take a look at the typical components in an API management system and see how they potentially connect to various parts of an organization.

API PUBLISHER

The API publisher is where the API developers design and publish APIs for consumption by external or internal consumers. This lets API developers control the lifecycle of APIs — from getting an API in design state to prototyped, available in production, and deprecated. It also enables the publishing of newer API versions, creating API products, and attaching documents to APIs. Let's take a look at some common integrations and required customizations to make the publisher fit in with an organization's regulations and policies.

USER REPOSITORIES

The user persona that performs these operations in most cases is internal to an organization. The API publisher, therefore, needs to connect to organization-specific user repositories and needs to have strict access control and governance rules to ensure that only the right set of people are given access to perform these actions. The software should have the capability to connect to any type of standard or proprietary user repositories.

API LIFECYCLE

All API management systems come with a default lifecycle for their APIs. The lifecycle of an API, however, can be specific to each organization that deploys the solution. Organizations need to have full control of the stages of their APIs and governance over what happens when an API changes its state. It is, therefore, necessary for the API lifecycle to be customizable to cater to organization-specific requirements. Most of these requirements are bound to customizable workflows that deal with user approvals, conformance checks, and so on. The solution should be able to integrate with standard business process modeling tools, using specifications such as Business Process Model and Notation (BPMN).

API POLICIES

API policies are used to perform various types of actions when serving API requests from clients. These can be related to logging some API information, including additional parameters to be sent to backend systems, excluding certain sections from messages, combining multiple services to orchestrate a single API flow, and so on. Many API users require common rules to be executed when serving API requests. These include passing a security token to a backend service and setting session attributes to requests. This results in many API users implementing their own API policies that effectively perform the same function. With an open source community, these types of policies can be easily shared among the user community. This prevents users from reinventing the wheel and helps them to get their tasks accomplished faster. It is even better if the API management system itself provides a formal mechanism of sharing API policies publicly.

API PORTALS

The API publisher component in an API management system pushes APIs' information to API portals. This is where these APIs are discovered for consumption. Organizations may prefer to make their APIs discoverable on their own portal solutions, which are already in use by other systems. This makes it necessary for the API publisher to integrate with third-party portal solutions. While these types of integrations mostly happen using APIs exposed by the respective portals, the API publisher requires exposing the necessary extension points for users to be able to create the integration.

APPLICATION DEVELOPER PORTAL

An app developer portal is an integral part of an organization's API ecosystem. The app developer portal is what makes your APIs discoverable and available to their consumers. Developers writing apps that consume your APIs would sign in to the application developer portal to discover, learn, and test APIs, create various types of applications, and subscribe to APIs through these applications.

Developers who sign-in to this portal could be either external (public) or internal users. Similar to that of API publisher user repositories, this requires the application developer portal to be integrated with different types of user repositories, which could be either standard or proprietary repositories. An open-source API management system makes such integrations easier.

In many cases, we now need capabilities that enable portal users to use their social network accounts to login. Almost all social network providers offer integrations over standard specifications, such as OpenID Connect (OIDC), OAuth2.0, and SAML. The application developer portal login flows being developed on standard protocols would make these types of integrations seamless.

We would also need to control/govern the actions performed by various users on the app developer portal. Similar to that of the API lifecycle, we would need to integrate the app developer portal with business processing modeling tools for better control and governance. These types of integrations make it easier to enforce organizational policies, such as multi-level approvals and conformance checks.

API GATEWAY

The API gateway is the main entry point into our business APIs. The API gateway intercepts all requests from client applications and performs various types of checks, such as authentication, authorization, rate limiting, policy enforcement, and analytics, before these requests are handed over to the backend business APIs. As such, the API gateway needs to integrate with multiple systems for serving APIs.

To perform security checks on APIs, the API gateway needs to integrate with the key server. In certain cases, organizations prefer to use a key server that is not from the API management vendor themselves. This requires the API gateway to be integrated with third-party/custom key servers. It is, therefore, essential for an API gateway to be customizable and able to work with third-party key servers over standard or proprietary protocols.

Similarly, the API gateway needs to have capabilities that allow it to integrate with legacy business services, observability tools, log analysis solutions, serverless functions, cloud services, and more. It is unlikely one would need all these types of integrations at the very beginning of a project. However, as time goes on and the project becomes more successful and popular, some, if not all of these integrations, will be necessary.

KEY SERVER

The key server acts as the main security gateway to your APIs. It is responsible for issuing and validating tokens (keys) for APIs. These keys are usually offered through standard APIs built on well-known specifications. In the context of OAuth2.0 (which is the de-facto standard for API security), these mechanisms are known as grant types. While there are well-known grant types for issuing tokens, not all client applications are capable of adhering to the protocols defined in the specifications, owing to certain limitations and organizational policies. It is, therefore, necessary to customize and extend these grant types to cater to such requirements.

Similarly, the key server, in most cases, is also expected to federate user authentication with other (third-party) identity and access management solutions. This could be due to organizational policies or it could even happen following business mergers and acquisitions. Having a key server that is able to perform such capabilities through open standards becomes highly beneficial in such cases.

Similar to most of the examples provided above, there are many cases where an API management solution needs to integrate with various other third-party systems to facilitate organizational requirements. This is where extensibility, customizability, and open standards play a critical role. Most API projects that start small will not have such needs at the beginning, and it is, therefore, easy to get carried away by choosing a solution that seems simple and inflexible. The prudent thing to do is to pick a solution that is simple enough, but also flexible enough, to cater to your requirements. Pay attention to the building blocks of the solution and pay attention to its details. Don't get carried away by the colorful wrapping. The building blocks are the ones that will make you powerful and agile enough to create a market-leading solution.

The Open Source Business

The new technology world has embraced open source. Almost every vendor now has at least some part of its offerings as open source. This is a strong statement that in the future we are only going to see increased adoption of open-source technologies.

Unlike enterprise licensing-based business deals, an open source agreement represents the beginning of a relationship between the two organizations. This is a deal/relationship that goes beyond a handshake. With open source vendors offering their products freely for use, the commercial aspects only become relevant when it comes to support and services. For an organization to obtain support from a vendor, the solution itself needs to be an important part of the organization's business. It is unlikely anyone would be paying for it otherwise. This simply means the success of the open source vendor solely relies on the sustained success of the organization obtaining support from the vendor. This sends a strong message to the world regarding the level of confidence the vendor has in the technology it offers.

Written by **Nuwan Dias,** Senior Director, WSO2

Nuwan is a part of the architecture team that drives development efforts of WSO2's API management platform. Previously, he spearheaded the platform's architectural and research and development aspects. He has also been involved in many customer engagements where he has assisted in developing solutions architectures.

Nuwan joined WSO2 in 2011 as a Software Engineer and worked mainly on the WSO2 ESB. He joined the WSO2 API Manager team in 2012 and has been an API enthusiast since. His interests lie in the enterprise integration patterns and API management domains. His other areas of interest include Java EE application design and development, SOA, and enterprise application security.

DZone® A DEVADA MEDIA PROPERTY

DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects, and decision makers. DZone offers something for everyone, including news, tutorials, cheat sheets, research guides, feature articles, source code, and more. "DZone is a developer's dream," says PC Magazine.

Devada, Inc.
600 Park Offices Drive
Suite 150
Research Triangle Park, NC

888.678.0399 919.678.0300