Chasing Unicorns and Clouds:

Managing APIs and microservices

01

Introduction

APIs and microservices are maturing, with implications for their security and management. APIs have evolved from scattered and selective efforts to become fundamental building blocks of enterprise architecture—with big increases in the scope and numbers of APIs that require management. Microservices have graduated from the proof-of-concept stage of the technology lifecycle.

Enterprises now expect to build reliable, critical applications using microservices. These changes create a need for far more integrated, automated and pervasive management for APIs and microservices. The new requirements for API and microservice management must address multi-cloud deployment, CI/CD nativity, ISTIO compatibility and more.

Understanding the business climate for APIs and microservices

Businesses face an accelerating pace of change and technological disruption. They want to transform themselves, offering customers a mobile experience, omni-channel relationships and beyond. We all know this. What's not so obvious to everyone, however, is how challenging it can be for IT to deliver the innovation needed to thrive in a disruptive world.

The race to digital transformation invariably requires deep, secure integration between applications housed in multiple corporate entities and hosting environments. Transformation also demands a software development cycle far more rapid than anything that came before.

To succeed, IT must balance agility, cost, responsiveness and security. Limitations in software architecture get in the way. Leading internet innovators, the 'unicorns' that change the world and reach new heights of business success, have shown that APIs and microservices provide an answer. They enable IT to innovate, with streamlined application integration and rapid, agile development taking advantage of deployment flexibility in the cloud, on premises or in a hybrid combination of the two.

How technology adapts to the business climate

Businesses seeking a transformation want to create exceptional experiences for customers and partners. They achieve this goal by enabling business capabilities and data to new modes of use. The results include innovative user experiences across channels and devices. Data analysis underpins it all—helping both IT and business stakeholders anticipate future needs.

This quest has been the story of IT over the last 20 years. A series of industry-wide initiatives has tried to enable data and business capabilities through new usage patterns. The Service-Oriented Architecture (SOA), for example, was built using standards-based SOAP Web Services and IP protocols. It succeeded in opening up many previously monolithic, proprietary and tightly-coupled systems to integration. Though revolutionary in its own way, SOA proved too complex to implement broadly.

APIs, and now microservices, are realizing the vision first espoused by SOA advocates. Through standards and universal norms of use, they make it possible to securely unlock data and expand application ecosystems with relative ease. The IT organization can manage the full API life-cycle to create, consume and control APIs. They can manage API policies at a granular level to optimize data access. Now, access is available to a vast array of enterprise files from any device anywhere, anytime regardless of where the files are stored.

Addressing challenges in API and microservices management

API and microservices enable businesses to build secure digital ecosystems of partners, suppliers and customers. The standards-based connections between systems make it possible for companies to easily onboard new partners, build APIs and devise partner API programs that harness external innovation. However, there have been some growing pains on the way to realizing the vision of the API- and microservice-powered enterprise. To some extent, APIs and microservices have been victims of their own success. Scale and complexity come with broad adoption. As each challenge has arisen, though, a solution has also emerged.

How API and microservices management solutions work today

As APIs and microservices have grown more prevalent, platforms have evolved to keep up with the complexity and scale of their management. Though they're still generally referred to as "API Management Platforms," in reality they do much more than management. *Figure 1* captures the multi-faceted nature of the new generation of API and microservice management platforms.

The most effective API and microservices management platforms in use today offer features to help with the use of APIs and microservices in application development, B2B use cases and content collaboration, e.g. between partners in an API community. They enable advanced analytics of API and microservice performance and dependencies. Additional features like Managed File Transfer (MFT) help in application development.
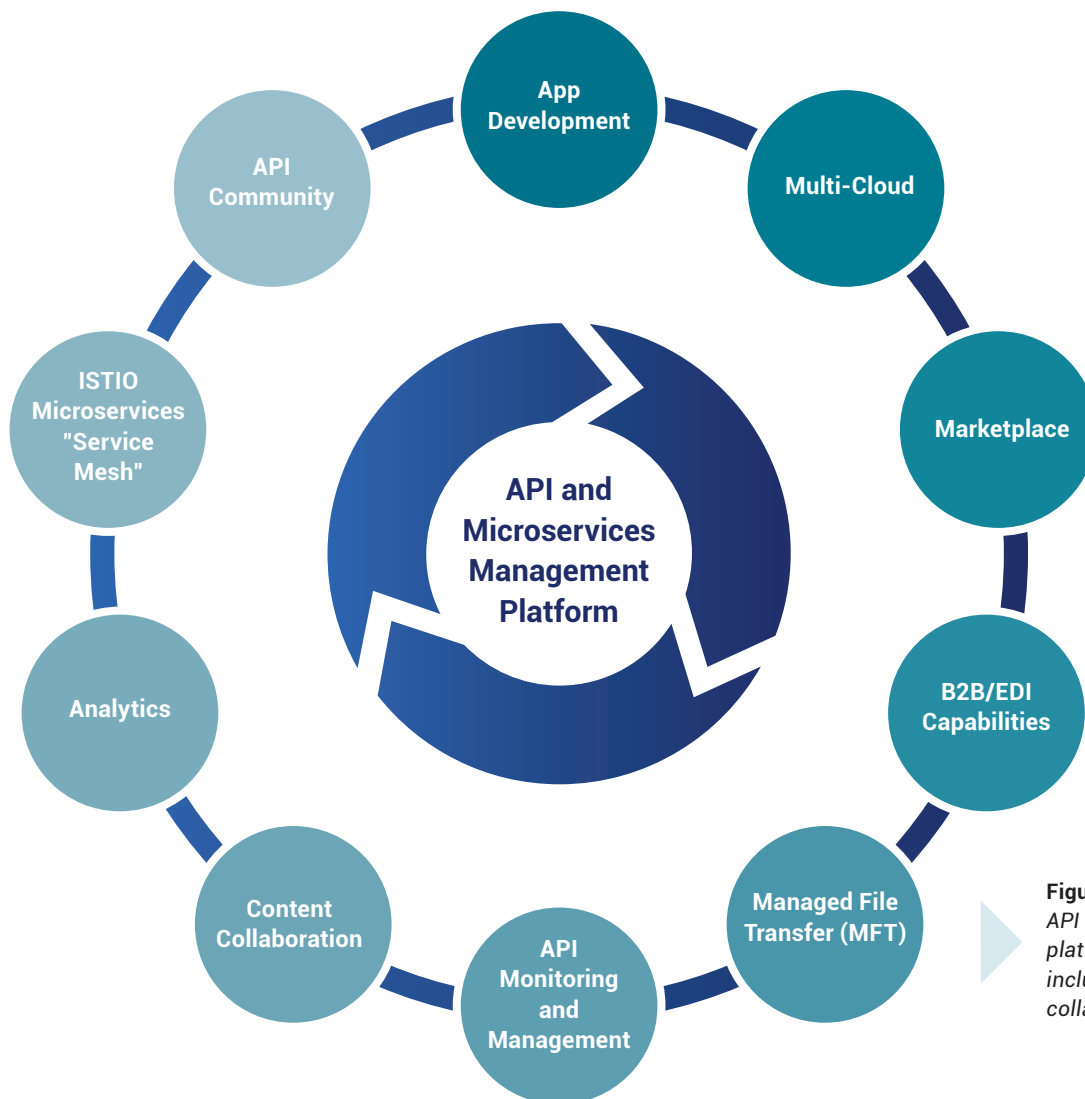

Figure 1: *The new generation of API and microservices management platforms span a wide range of features, including application development, content collaboration, analytics, ISTIO and more.*

Getting APIs and microservices to work at true enterprise scale

API and microservice management solutions contribute to solving a seemingly simple but serious challenge in the modern API-driven enterprise: the pure scale of adoption. Once a specialized form of application integration, APIs and microservices are now appearing in virtually all areas of the application landscape. Packaged applications now ship with ready-to-go APIs. Numerous web- and cloud-based digital services are built with APIs as the only point of potential connection. The majority of new, in-house application development efforts assume API connectivity as well as a functioning or potential microservice architecture.

Such widespread use of APIs and microservices creates complexity in the architecture. With applications connected—and regularly reconnected again and again—it's easy to lose track of the points of integration. Nested dependencies can wreak havoc on the performance of connected applications. Microservices skills, once the province of a minority of experts, must now be practiced by a far bigger group of IT professionals. All of this puts a strain on the ability of the enterprise to keep the API - and microservices-based architecture functioning.

Moving from isolated teams to organization-wide embrace

A management platform contributes to solving the scale problem by making organization-wide API and Microservice use a possibility. With a common user interface, a community, global API catalog and API "Marketplace," the new generation of API management platform is inviting and more usable than its comparatively esoteric predecessors. It facilitates a move from isolated teams supporting APIs to the organization-wide embrace and use of APIs and microservices that are independently versionable, and scalable.

Eliminating "wizards" and the "tall enough to ride" phenomenon

Microservices skill sets, once rare and arcane, must now become more accessible to a much broader range of people.

As a specialty, microservices were a bit nerdy, the domain of "wizards" who could be counted on to perform impressive tricks of development and integration.

A microservices project was like an amusement park ride with a "You must be this tall to ride" sign at its gate. As long as the company, organization or team was "tall enough," in skills and organizational terms, they could do the work and establish their microservice strategy with success.

This is fine as long as your organization is viewed as a technology innovator like Netflix. However, most enterprises on a budget can't afford the time or the microservices experts that can assemble all the open source components needed. Nor do they generally have the ability to establish the DevOps CI/CD pipeline with ease. Now that these microservices technologies are omnipresent and the benefits widely understood, intuitive management tools can make every company "tall enough" and full of wizards, so to speak.

Distributing control through the organization

Architectural independence is one of the great values of APIs and microservices. Standards-based and self-contained, they do not need centralized control. This is good and bad. Uncontrolled, APIs and microservices can spawn chaos, affecting application performance and systemic reliability. With excessively centralized control, the flexible essence of the technology gets stymied. New API and microservice management platforms resolve this tension with self-service provisioning. This distributes control through the organization.

The platform gives relevant stakeholders the visibility they need to keep all elements of the architecture running smoothly. At the same time, users can manage their own APIs, ensuring agility overall.

Adopting microservice management systems (ISTIO)

Microservices break down applications into small component services. This makes them extremely flexible and adaptable. At the same time, it can be a headache to manage them. There are so many of them! They are all over the place! In response, the industry has come together under ISTIO, an open source, open platform to connect, manage and secure microservices.

ISTIO creates what is known as a 'service mesh' that unifies microservices traffic flow management. It centralizes microservices access policy enforcement and aggregates microservices telemetry data via a shared management console. The platform works with containers like Docker, which have become standard for cloud deployments of microservices. Containers, while useful and necessary, bring their own complexity to developers who want to orchestrate microservices throughout a heterogeneous architecture.

Streamlining orchestration

Figure 2 shows a mesh app and service architecture and the kind of complex dependencies that can arise when developers and architecture orchestrate multiple APIs and microservices across several different applications. When APIs first appeared in enterprise architecture as Edge APIs, and specified the "North to South" flow of information—meaning, the API invocation and response went up and down the application stack from consumer to service.

This is less and less the case today. Apps now routinely invoke APIs which will call microservices that exist elsewhere in the architecture. They represent an "East-West Traffic" or service-to-service flow of information. This enable services to speak directly to each other through micro gateways instead of having them all tunnel through a single edge

gateway. The east-west mode of traffic improves agility and resiliency.

The API and microservices management platform can set up an API orchestration and a mediation layer between the Apps, edge APIs and connected microservices. The platform provides a Service Mesh, typically through ISTIO, to enable reliable, monitored East-West interactions between microservices amidst multiple applications.

The platform also makes possible a streamlined orchestration process. Given the kind of dependencies shown in *Figure 2*, this is very much needed in today's architecture. Locating microservices, making the connection and then monitoring the results takes a powerful management platform.
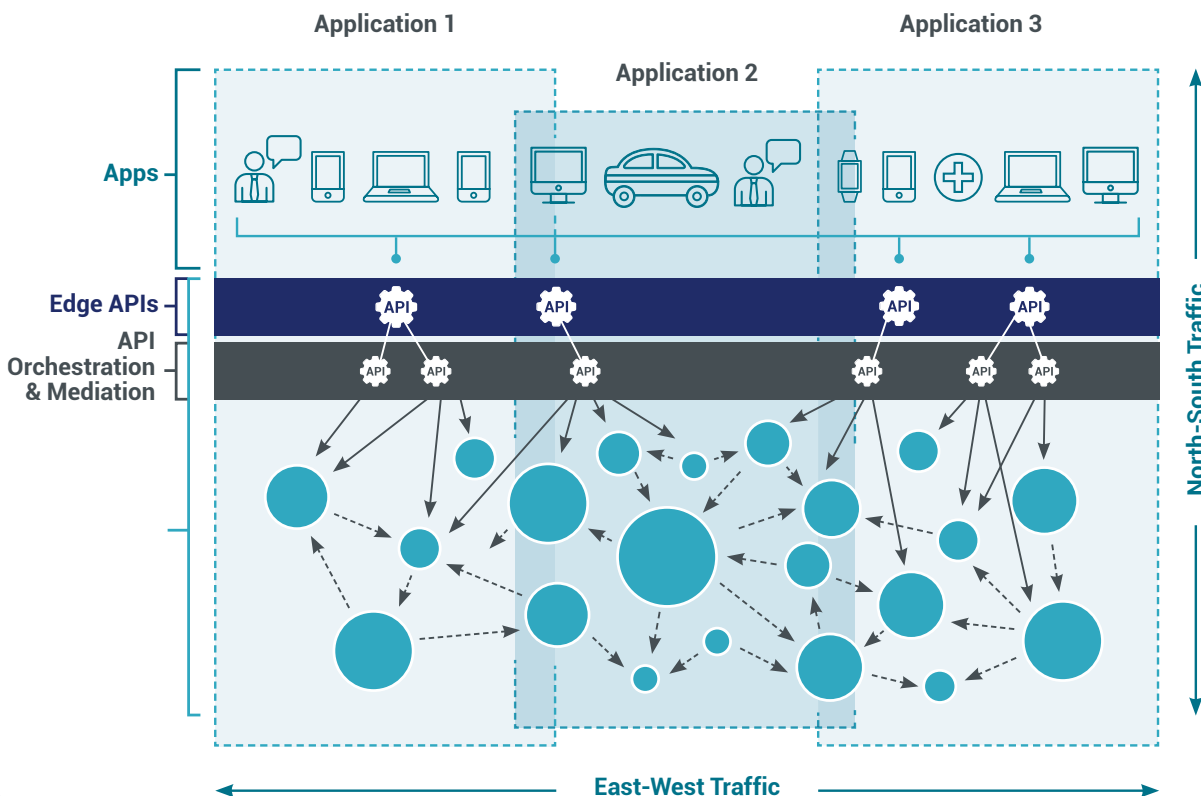

Figure 2: *Microservices and API orchestration and dependencies can create highly complex performance problems that are challenging to solve. Without the right diagnostic and monitoring tools.*

Putting an end to "murder mysteries"

The environment portrayed in Figure 2 can make for "fun" experiences detecting the source of a performance problem. Sometimes called "murder mysteries" by API admins, the process of discovering and remediating a problematic microservice can be stressful and wasteful of resources. The new generation of management platforms provides a higher level of transparency and discoverability. In this way, it puts an end to the murder mysteries and allows API admins to remediate performance issues quickly.

Elevating microservices management

It's common for people to refer to "APIs and microservices" as if they were an organic pairing. This makes sense, given the similarity and synergy between the two technologies. However, in terms of management, microservices can suffer from second-class citizen status.

A good practice is to elevate microservice management to the same status as API management.

Then, it becomes easier to support high-volume, enterprise-scale use with the ability to manage traffic in all directions.

Enabling API-and-microservice-driven software development at scale

APIs and microservices must fit naturally into the software development process. After all, much of what constitutes a modern software application is actually hosted elsewhere and accessible through APIs. Apps contain APIs, too. Some apps are simply made of many microservices and a few edge APIs, For this to work well, it's necessary to align API and microservices management solutions with DevOps. Figure 3 shows what this looks like.

The DevOps cycle, shown in the figure as an infinite loop, is indeed a never-ending flow of planning, building, testing releasing, deployment, operating and so forth. As APIs and microservices attach to applications as they take shape, the API management platform has work to do at each stage of the DevOps cycle. API review takes place at the planning stage. Coding and building match with API and Mmcroservice creation. API testing aligns with app testing. App deployment requires API configuration management.

One of the most critical "must haves" in making APIs and microservices work with DevOps is the ability to make APIs and microservices native for Continuous Integration (CI) and Continuous Deployment (CD). Code today is released constantly, often right into production. APIs and microservices have to snap to the organization's environment quickly and enable continuous change.
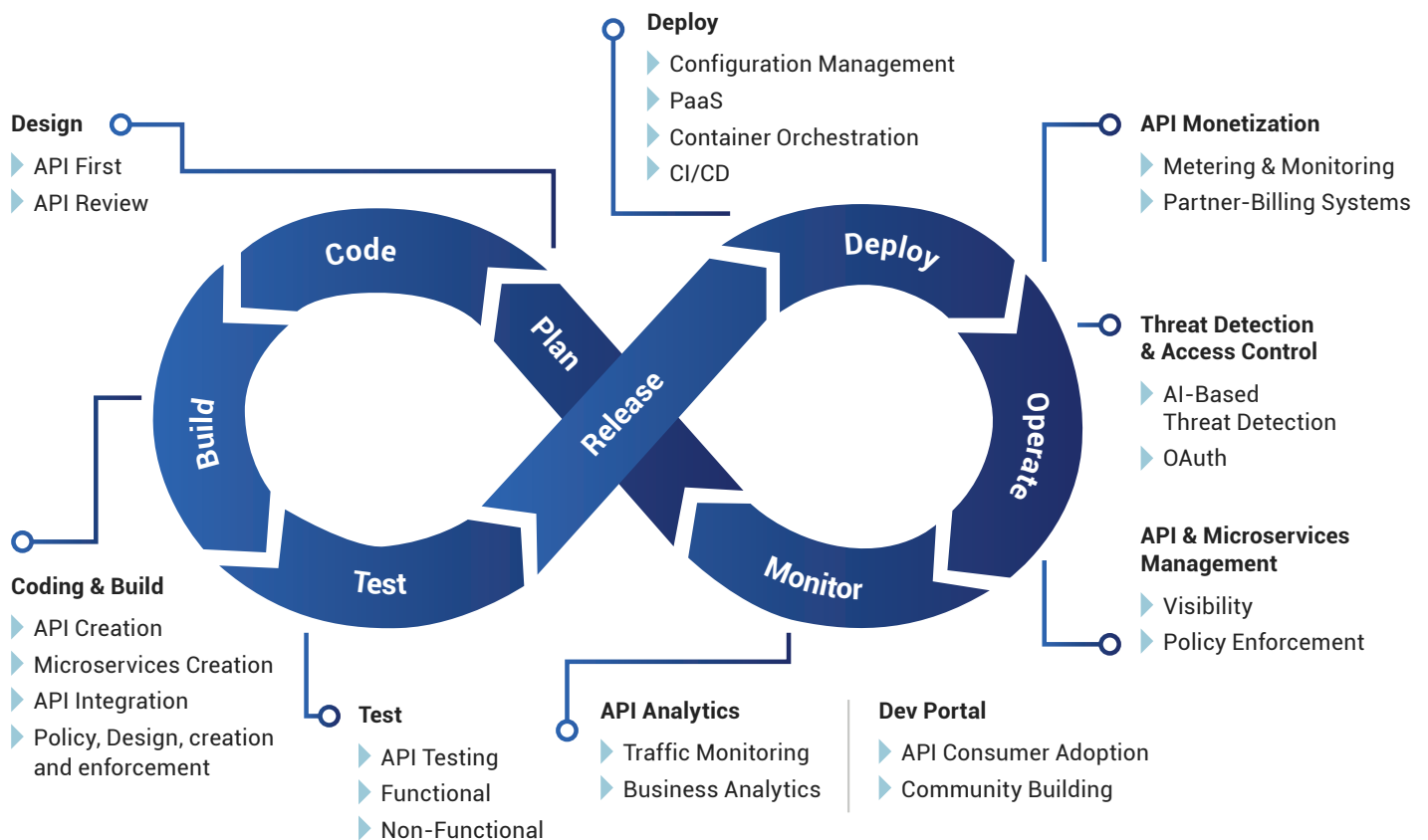
Deploy
- Configuration Management
- PaaS
- Container Orchestration
- CI/CD

Design
- API First
- API Review

API Monetization
- Metering & Monitoring
- Partner-Billing Systems

Threat Detection & Access Control
- AI-Based Threat Detection
- OAuth

Coding & Build
- API Creation
- Microservices Creation
- API Integration
- Policy, Design, creation and enforcement

API & Microservices Management
- Visibility
- Policy Enforcement

Test
- API Testing
- Functional
- Non-Functional

API Analytics
- Traffic Monitoring
- Business Analytics

Dev Portal
- API Consumer Adoption
- Community Building

Figure 3: *Alignment between the DevOps cycle and API and Microservices management*

Deploying in multi-cloud

APIs and microservices are now deployed in every imaginable hosting environment, They're on-premises, in private clouds, public clouds and hybrid clouds. For APIs and microservices to work at enterprise scale, management tools have to be able to work across multi-cloud and hybrid deployments.

This way, an enterprise can access APIs and microservices no matter where they're located— and do so with a unified means of monitoring, managing and securing them.

Conclusion

The management platform makes a big difference in the ability of the enterprise to put APIs and microservices to work. They enable a bigger set of users to take advantage of APIs and microservices at scale. This way, they can leverage their connectivity and flexibility to innovate and move the organization toward digital transformation. For APIs and microservices to work at scale, the management tool must have certain capabilities.

They must be integrated, automated and pervasive. The management platform has to support the ISTIO service mesh, multi-cloud deployment, DevOps and CI/CD nativity. With these capabilities, APIs and microservices can serve as the foundation for exciting new experiences for customers and partners.

Try AMPLIFY API Management online for free

SIGN UP TODAY

axway.com/en/products/api-management