apigee

Google Cloud

Maximizing Microservices

Tame complexity and extend value with API management and a service mesh

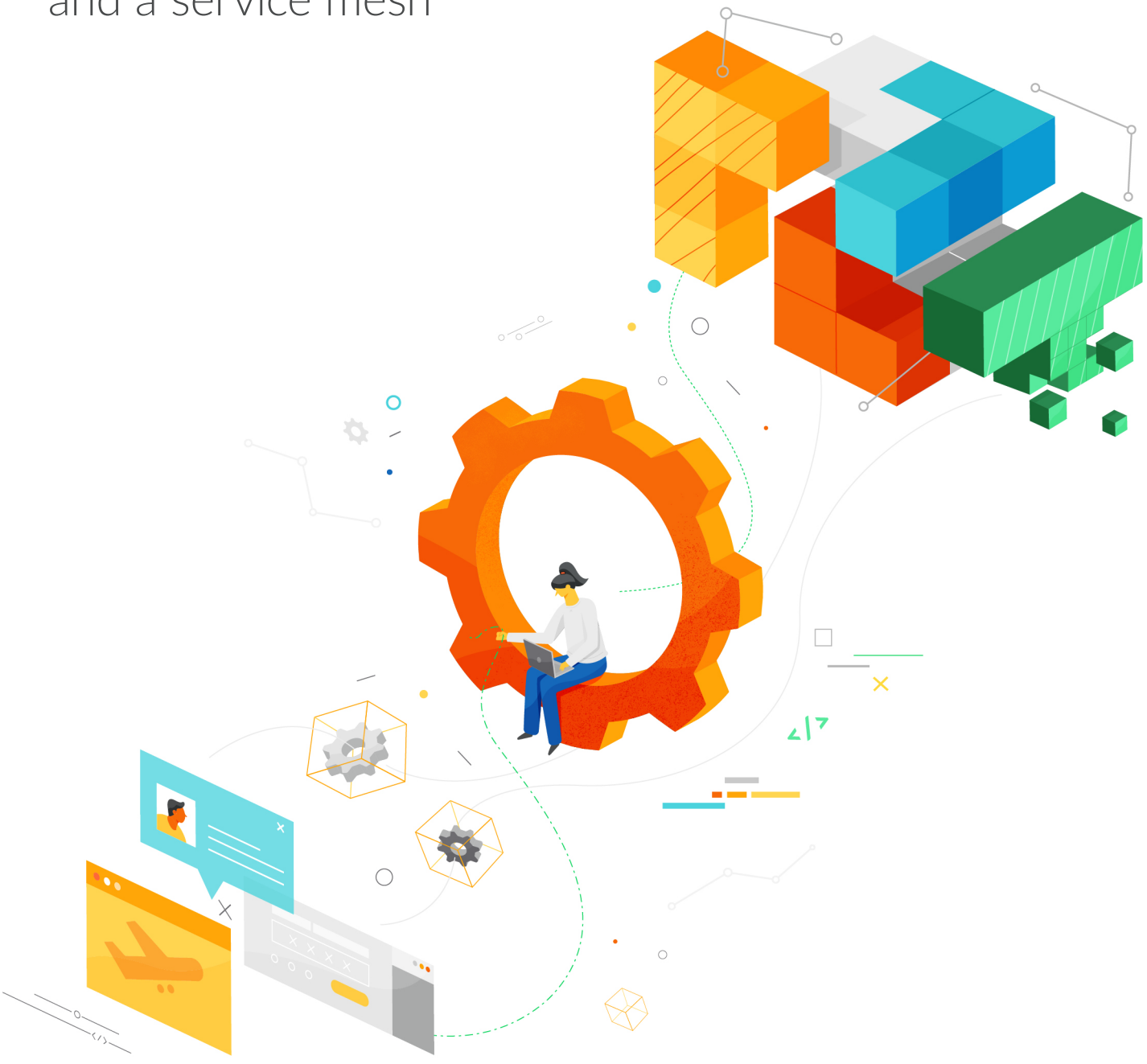


Table of contents

Executive summary 03

More microservices mean more complexity 05

Shared microservices are packaged and managed as APIs 07

How it all fits together:
Understanding the microservices management stack 08

APIs and microservices in action 11

- How PwC Australia leverages microservices and APIs for new lines of business 11
- How Magazine Luiza soared with microservices and APIs 12

Remember—it's all about customers 14

Executive summary

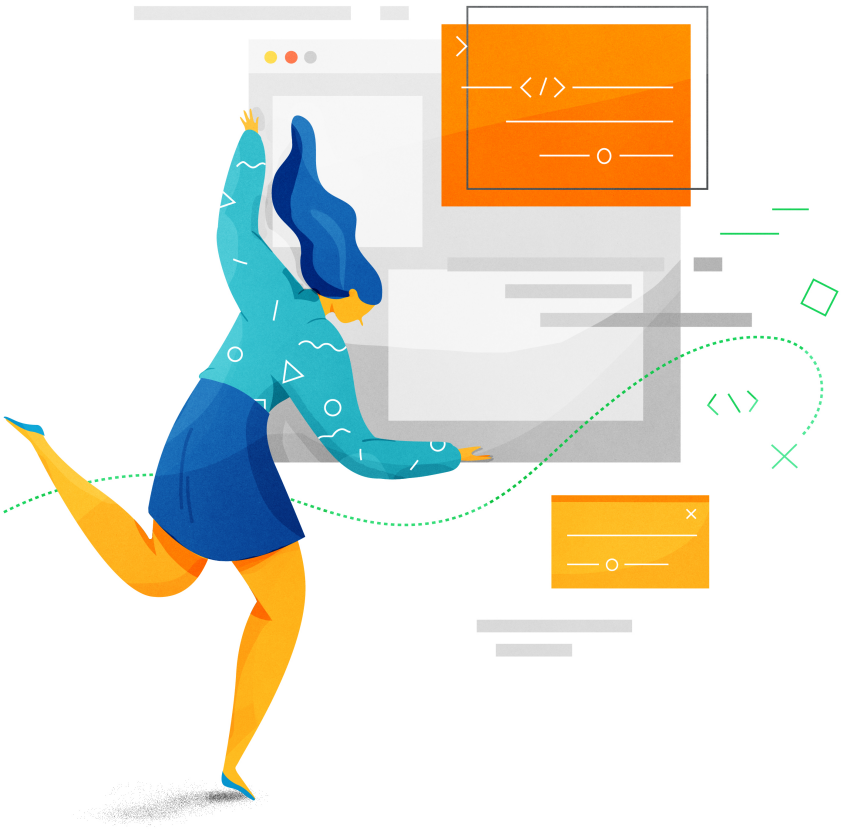
Over the last few years, microservices architectures have been increasingly celebrated as a way for enterprises to become more agile, move faster, and deliver applications that keep pace with changing customer needs.

A microservices approach is a significant departure from traditional software development models in which applications are built and deployed in monolithic blocks of tightly coupled code. These legacy approaches can make updating applications time- consuming, increase the potential for updates to cause bugs, and often limit how easily and quickly an organization can share or monetize its data, functions, and applications. Microservices, in contrast, are fine-grained, single-function component services that can be scaled and deployed independently, enabling organizations to update or add new features to an application without necessarily affecting the rest of the application's functionality.

Because of their autonomous and atomic nature, microservices can help a business achieve unprecedented levels of agility, empowering development teams to innovate faster by building new features and services in parallel. But these benefits come with some costs.

Managing the complexity of large numbers of microservices can be a serious challenge; doing so demands empowering developers to focus on what microservices do rather than how they are doing it. For this, enterprises are increasingly using a "service mesh"—an abstraction layer that provides a uniform way to connect, secure, monitor, and manage microservices.

A network of microservices is complex—even if the services are used within a single team. This complexity compounds when enterprises want to increase consumption and extend the value of microservices throughout the organization or with partners and external developers. For this, an enterprise needs managed application programming interfaces (APIs). APIs and API management help expand the universe of developers who can take advantage of microservices, while giving organizations governance over how their microservices are used and shared. Whenever a microservice is shared outside the team that created it, that microservice should be packaged and managed as an API.



Put simply, if an enterprise is serious about its microservices strategy, it needs both a service mesh to help simplify the complexity of a network of microservices and API management to increase consumption and extend the value of microservices to new collaborators. A microservices strategy that lacks either of these elements will likely struggle to gain momentum, let alone scale beyond bespoke projects.

In this eBook, we explore:

- The role of a service mesh in simplifying complexity intrinsic to microservices architectures
- How APIs enable the value of microservices to be scaled and shared with additional teams, developers, and partners
- Why an enterprise's ability to secure, monitor the use of, and derive insights from microservices relies on properly managing the APIs that make microservices accessible
- How a comprehensive microservices strategy combines both a service mesh and API management to manage complexity and securely increase consumption

More microservices mean more complexity

Small, fine-grained functions that can be independently scaled and deployed, microservices provide software development teams with a new, agile way of building applications.

As microservices architectures have become more closely associated with enterprise agility, microservices investments have accelerated across the business spectrum—not just among big companies, a majority of which[1] are either experimenting with microservices or using them in production, but also among mid-market firms and SMBs.[2]

Given the success stories that have accumulated, it's easy to understand the enthusiasm. Netflix's iterative transition from a monolith to micro-services[3] has famously helped the company to make its content available[4] on a dizzying variety of screen sizes and device types. South American retailer and Google Cloud customer Magazine Luiza has similarly leveraged microservices to accelerate the launch of new services, from in-store apps for employees to an Uber-like service to speed up deliveries, and help it earn praise as the "Amazon of Brazil".[5] Other major microservices adopters, including Airbnb, Disney, Dropbox, Goldman Sachs, and Twitter, have cut development time significantly.[6]

---

[1] DZone, "DZone Research: Microservices Priorities and Trends" by Anne Marie Glen, July 2018.
https://dzone.com/articles/dzone-research-microservices-priorities-and-trends

[2] CRN, "Research: CIOs Up Spending On Containers, Microservices As Companies Increase Public Cloud Use" by Alec Shirkey, September 2017.
https://www.crn.com/news/cloud/300092736/research-cios-up-spending-on-containers-microservices-as-companies-increase-public-cloud-use.htm?

[3] Netflix, "Engineering Trade-Offs and The Netflix API Re-Architecture" by Katarina Probst and Justin Becker, August 2016.
https://medium.com/netflix-techblog/engineering-trade-offs-and-the-netflix-api-re-architecture-64f122b277dd

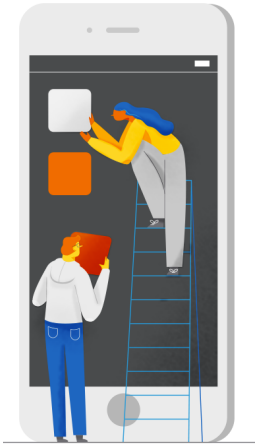
[4] re/-fraction, "How Netflix works: the (hugely simplified) complex stuff that happens every time you hit Play" by Mayukh Nair, October 2017.
https://medium.com/refraction-tech-everything/how-netflix-works-the-hugely-simplified-complex-stuff-that-happens-every-time-you-hit-play-3a40c9be254b

[5] Bloomberg Businessweek, "Hawking TVs on Tinder Helps Fuel 2000% Rally in Brazil Stock" by Fabiola Moura and Paula Sambo, August 2017.
https://www.bloomberg.com/news/articles/2017-08-14/hocking-tvs-on-tinder-helps-fuel-2000-rally-for-brazil-retailer

[6] Computerworld, "How to determine when and why to use microservices" by Gary Olliffe, June 2017.
https://www.computerworld.com.au/article/621169/how-determine-when-why-use-microservices/

It's clear that when microservices are implemented and managed well, they can deliver new levels of scale, speed, and responsiveness—the major IT ingredients a company needs to compete and delight customers.

Implementing microservices successfully is notoriously complicated, however.



Instead of deploying all the code each time the application is updated, as is common in monolithic application archi- tectures, enterprises can leverage microservices to deploy different pieces of an application on different schedules. For this to work, individual teams or developers need the freedom to refactor and recombine services based on how the larger application is consumed by users. Because a microservice in an application depends on all the other micro- services that compose the application, this complexity needs to be abstracted and managed so that one team's work doesn't break another's.

If a business fails to recognize that complexity increases with the number of microservices it uses, the organization's efforts are unlikely to succeed. Martin Fowler, one of the intellectual authors of the microservices movement, highlights[7] "operational complexity" as one of the key drawbacks of the approach, and Gartner research vice president Gary Olliffe has warned that a majority of enterprises may find "microservices too complex, expensive, and disruptive to deliver a return on the investment required."[8]

As the use cases for microservices have expanded, so has the complexity. The original vision of microservices held that a microservice wouldn't be shared outside the team its creator worked with. Among the things that made them "microservices," as opposed to APIs or service oriented architecture (SOA), was the fact that developers no longer had to worry about the same level of documentation or change management that they did with a widely shared service.

But microservices have become heralded as a valuable way to reuse functions and scale them to more developers, both inside and outside an organization. The granularity and agility they provide is too valuable to confine within a single team. As enterprises have attempted to extend the value of microservices to more teams and partners, many businesses have struggled to make microservices secure, understand how microservices are used and are performing, and successfully deploy microservices beyond bespoke use cases.

[7] MartinFowler.com, "Microservice Trade-Offs," July 2015. https://www.martinfowler.com/articles/microservice-trade-offs.html

[8] Ibid, Computerworld

Shared microservices are packaged and managed as APIs

Microservices present management challenges from two angles—the complexity that arises from a growing network of microservices and the intricacies that result from sharing microservices as APIs with new teams and developers.

To conquer the first challenge, enterprises must recognize that as they increase the number of microservices in production, the complexity of managing service-to-service communications within even a single team can dramatically increase. It's important that developers be able to focus on the functionality microservices provide rather managing the complexities of how they interact. For this, businesses are increasingly using a service mesh such as the open source Istio project. A service mesh provides a uniform way to connect, secure, manage, and monitor microservices without forcing developers to (probably inconsistently) bake these features into their service code.

Moving to the second challenge, as enterprises share microservices, they need APIs to package them for easy developer consumption. Any time a microservice is shared outside the team that created it, the microservice should be presented as an API.

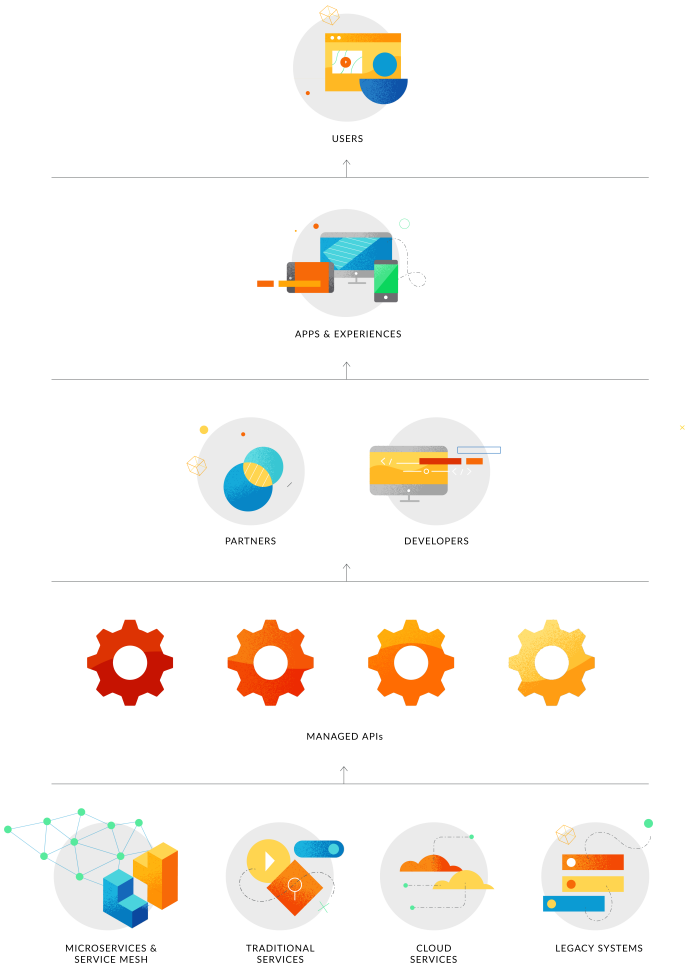
All APIs need to be managed. Without management, an organization can't gauge how reliable its systems are and how developers are adopting APIs and microservices. Without API management, the organization cannot determine how easy it is for developers to consume APIs, control who consumes APIs, and dictate how much traffic each API consumer uses. The organization has no assurance developers are implementing security precautions properly—or at all. When a business wants to scale microservices to new teams, partners, and developers, API management must become a cornerstone of its strategy.

Doing microservices well, then, means two things: applying a service mesh to maintain resilience and security while freeing developers from having to implement these solutions into their code; and using well-managed APIs to extend the value of these microservices beyond the teams in which they were created.

How it all fits together: Understanding the microservices management stack

Microservices are typically deployed in containers such as Docker that provide everything needed for the service to run. Containers are a significant architectural departure compared to legacy applications that ran on purpose-configured hardware, and the agility they can provide is one of the key reasons microservices can accelerate multi-cloud strategies; microservices can dynamically scale up or down the resources they need and apps can draw from services spread across many clouds.

"We can run our workloads anywhere. Microservices and Kubernetes give us freedom. It's been especially helpful with our multi-cloud strategy," Maga- zine Luiza CTO André Fatala said in an interview.



But the very reasons for the success of this architectural model also present some challenges. Today's developers build APIs and microservices without the kind of centralized oversight that once existed. Because an application might rely on calls to many services, it can be an enormous challenge to manage which services are allowed to communicate and how calls should be routed to maintain excellent end user experiences.

In modern decentralized application architectures, containers offer the first important layer of control and resiliency. Typically, when enterprises deploy containers, they apply an orchestration layer such as Kubernetes to abstract the underlying hardware and enable the services to be exposed to developers via an API. The orchestration layer facilitates several important infrastructure scaling functions as well as transport layer load balancing and health checks.

A service mesh such as Istio constitutes the next layer in the microservices management stack. Its responsibilities include application layer load balancing, service authentication, policy enforcement, routing, telemetry reporting, and other important aspects of service-to-service control and reliability.

In essence, the service mesh lets developers decouple network functions from their service code. Developers don't need to implement code for these resiliency and management functions in their services—they can focus on what the service does rather than the complexities of how it communicates in the underlying network.

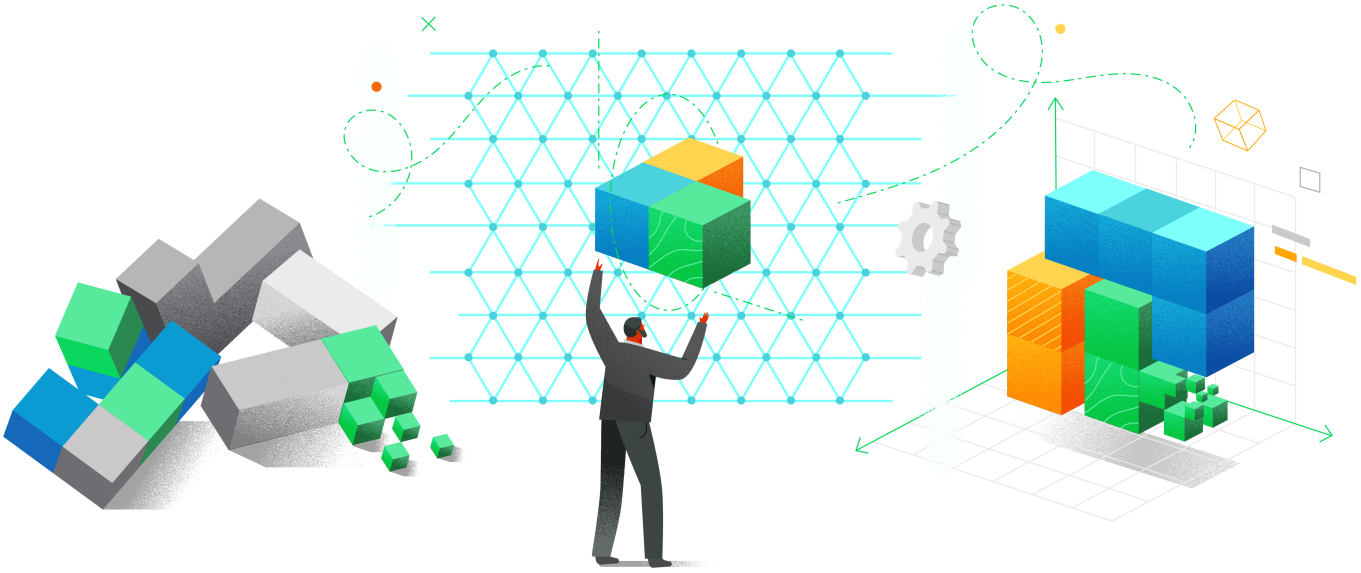
APIs sit above the service mesh and enable microservices to scale to more developers, inside and outside an organization. Though APIs are necessary to expose microservices, APIs and microservices are not the same. APIs can expose systems and digital assets beyond microservices, for example, and APIs support deeper levels of management functionality. API management is vital to enforcing policies, and potentially upholding service-level agreements (SLAs), around the use of those microservices.

As companies open up access to microservices and other digital assets via APIs, they must assume they are operating in a zero-trust environment. When developers deploy microservices in the public cloud and neglect to include common API security standards or consistent global policies, they expose the enterprise to potential security breaches. An API management platform enables enterprises to implement security and governance policies such as OAuth2 and bot detection across all of their microservices APIs. They also provide a plane for analytics and reporting, granting visibility into and control over how microservices and other digital assets are used.

API management helps companies not only secure their APIs but also make them more consumable and useful to developers. The service mesh typically includes a registry of microservices to facilitate service-to-service communication, for example, but the API management platform includes discoverability tools that help developers avoid re-creating APIs that are already available and so help the enterprise avoid development bloat. A robust API platform should also facilitate onboarding of developers via a self- service portal, include documentation resources, and offer tools for API monetization and productization.

"We want to expose more than just data [with our APIs]. We also want to expose functional elements for our developers to accelerate what they could create and imagine," Trent Lund, head of Google Cloud customer PwC Australia's Innovation and Ventures group, said in an interview.

Ultimately, a services management stack should include both a service mesh to keep microservices connected and secure while freeing the developers from service management distractions, and an API management platform to provide security, control, and visibility for all a company's APIs.



This relationship between a service mesh and API management is so important that some of today's most popular solutions have begun to bake aspects of both into their offerings. For example, Google Cloud's Apigee API management platform is now natively integrated with the Istio service mesh. This integration enables microservices to be easily exposed as APIs, while taking advantage of Apigee's robust API management capabilities.

APIs and microservices in action

CUSTOMER SPOTLIGHT

How PwC Australia leverages microservices and APIs for new lines of business

A part of "Big Four" accounting firm PricewaterhouseCoopers, PwC Australia is leveraging microservices and APIs to accelerate innovation and create new lines of business.

The organization has decades of experience in traditional professional services such as auditing, insurance, tax, legal, and management consulting—but over the last few years, PwC Australia's Innovation and Venture group has set out to redefine offerings for today's data-driven market, replacing backward-facing, reactive, and labor-intensive legacy approaches with more forward-looking, proactive digital methods.

Cash Flow Coach, for example, is a new product that applies machine learning models to ledger and banking data in order to predict cash flows based not only on when invoices should be paid but also when customers have traditionally paid. Rather than representing the modernization of an existing service, the product is an entirely new line of service, made possible by assembling data and functionality via APIs.

PwC's efforts also include a product that uses blockchain and microservices to prevent counterfeiting in the meat industry, such as old or sub-standard meat products whose fraudulent health and provenance information could be dangerous to consumers. The tool relies on a physical "krypto anchor" (an edible substance stamped on meat) that can be scanned at the point of unpacking in order to verify it matches a blockchain-based certificate. When it does, the meat's data is verified.

PwC is also packaging and monetizing services via APIs to open its data and technologies to new ecosystems of external partners and developers. Because these APIs are based on a microservices architecture, PwC can observe how its services are being used and responsively update particular aspects without disrupting the APIs developers use or breaking the end user experiences those APIs enable.

"That really is the key part of our strategy," said Trent Lund, head of PwC Australia's Innovation and Ventures group. "We act as a middle layer. Extract, but don't try to own the entire business ecosystem because you can't grow fast enough."

Part of a global development fund that includes PwC branches in New Zealand, the UK, and the US, PwC Australia's efforts have helped it improve development velocity by 30 percent and decrease DevOps costs by 20 percent.

CUSTOMER SPOTLIGHT
How Magazine Luiza soared with microservices and APIs

South American retailer Magazine Luiza is a strong testament to the results a company can achieve when it deploys microservices and APIs with vision and purpose. A traditional brick-and-mortar business for much of its history, the company has seen its stock become one of the hottest in Brazil[9] over the last few years and has enjoyed skyrocketing revenue as a focus on modernizing IT has become a bigger part of its approach to business.

In Q2 of 2018, Magazine Luiza's net profit grew 94 percent year-over-year, with e-commerce growing 66 percent and reaching a third of total sales. Less than a decade ago, however, Magazine Luiza appeared far less technologically imposing, showing little indication it could compete with fast-moving digital upstarts.

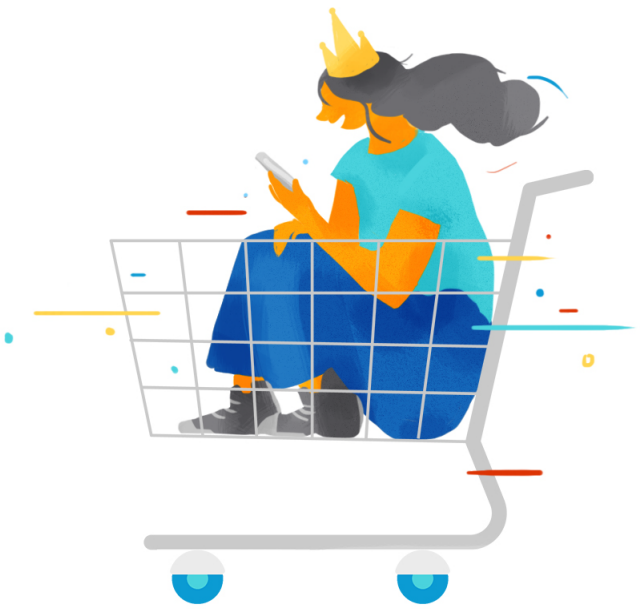
Back then, the company's e-commerce efforts relied on a monolithic backend built with over 150,000 lines of code. Its tightly-coupled, brittle nature slowed deployment of new APIs, created undesirable and often unanticipated dependencies that made updates difficult, reinforced silos between business and IT teams, and posed scalability challenges.

[9] Reuters, "Brazil's Magazine Luiza reports strong profit as e-commerce takes off" by Gram Slattery, May 2018. https://www.reuters.com/article/magazine-luiza-results/update-1-brazils-magazine-luiza-reports-strong-profit-as-e-commerce-takes-off-idUSL1N1SE216

A switch to microservices, pioneered with a small team of engineers before being expanded company-wide, removed many of these obstacles, helping the company to spin up new services and modify existing ones much more quickly. The company had been delivering only eight or so new software deployments monthly but has increased to more than 40 per day. Many small, five or six-people teams now work simultaneously on different services, from online checkout to order management to inventory.

The company was able to scale up its efforts, going from a handful of engineers to more than one hundred in just a few years, because executive leadership not only understood the company's digital vision but also enforced mandates to align the organization—such as using APIs as a communication interface between microservices and other systems.

A digital marketplace launched in 2016 is among Magazine Luiza's most noteworthy and transformative recent digital efforts. Even before the shift to microservices, Magazine Luiza had e-commerce capabilities but they were limited by a legacy system that supported fewer than 50,000 SKUs. The new marketplace, in contrast, enables new merchants to join via an API, enabling Magazine Luiza to not only to sell its own inventory but also partner at scale with sellers throughout the world. The new marketplace contains over 1.5 million SKUs and continues to grow.



Remember—it's all about customers

To get the most leverage from a microservices approach, a business needs both a service mesh to manage microservices networks and API management to maintain security, control, and visibility as microservices are extended as APIs to more partners, teams, and developers. With this combination, corporations are equipped to reduce the complexity that has hamstrung many microservices efforts, and to accelerate developer innovation by increasing consumption of valuable microservices via APIs.

But beyond the technology itself, it's important to remember microservices and APIs aren't just about scale, agility, or any other IT buzzword—they're about creating better experiences for customers. That's the reason microservices have become so popular. The number of new updates that a development team pushes, the number of services in a given compute cluster, and the number of developers consuming an API are all important—but only because these factors have helped companies to continue delighting customers.

Now that you've finished reading,
why stop learning?

MAXIMIZING MICROSERVICES MICROSITE

Take a deeper dive into the concepts and strategies you've learned to maximize microservices by exploring more success stories, videos, eBooks, articles, and more.

KEEP LEARNING

FREE TRIAL

Explore Apigee Edge, a full lifecycle API management platform that helps you manage the entire API lifecycle from design through iteration and helps you control the complexity of microservices.

TRY IT FREE

APIGEE COMPASS

Enterprises that take an API-first approach to digital transformation are poised to compete. Take a quick assessment and learn how your company stacks up against the core dimensions of digital transformation.

GET DIGITAL SCORE

apigee

Share this eBook

on social

with a colleague

Google Cloud