MICROSERVICES: ALWAYS AGILE. ALWAYS CURRENT.

How often do you have to patch, upgrade, and test your platform? How long are your systems down each time? If you're operating your business in a non-cloud environment, downtime is a reality. Maybe it's become such an acceptable part of your IT regime that you don't even question the restrictions and operational time lost. What if there was another way?

Imagine a world in which your software is always up to date and never needs downtime. Gone is the regular ebb and flow of upgrades, testing, and patches. There's no need for the seasonal, pre-peak trading lockdown. Say goodbye to unnecessary inefficiencies. That's what happens when you migrate to a cloud-native platform.

THE CLOUD'S SECRET INGREDIENT

You may be asking how it's possible to deliver IT services so seamlessly. The secret to an always-current, run-anywhere, cloud-native system lies in its microservices architecture.

The microservices approach is an increasingly popular way of building retail-specific and sector-agnostic enterprise systems. Essentially, it's a method of developing software applications as a suite of small, independently deployable, modular services, each running a unique process and communicating to serve an overall business objective through a well-defined mechanism.

In a retail environment, those goals could include enterprise order management, point of sale, inventory management, fulfillment, or customer intelligence.

To fully understand the attraction of microservices, it's useful to compare them with the traditional method of doing things: the monolithic architectural style.

CLOUD VERSUS MONOLITH

Traditionally, software was built as a single, autonomous unit. In a client-server model, the server-side application is a monolith that handles HTTP requests, executes logic, and retrieves/updates the data in the underlying database.

The problem with this type of architecture is that all change cycles usually end up being tied to one another. A modification made to a small section of any application might require building and deploying an entirely new version.

Rigorous and sometimes expensive testing regimes are often required — which is why risk-averse retail sector organizations play it safe, locking down their code weeks, if not months, ahead of peak trading. The kinds of restrictions associated with monolithic software can stifle agility, smother innovation, and kill new business opportunities before they can be realized.

For example, a company may be unable to add a hot new line to its inventory in the run-up to peak trading, while its agile, cloud-enabled competitors enjoy a critical advantage, capitalizing on customer trends to create new revenue opportunities

Retailers operating on a cloud platform can modify on the fly. This might include everything from initiating a simple workflow change in a mobile app, to wiring in an external promotions engine, to connecting API calls to a CRM application, quickly and easily.

Manhattan Associates®

While there's no single definition for microservices, they generally share a set of core characteristics.

They're modular – Microservices software can be disassembled into its individual functions. This means that specific components can be detached, upgraded, and then redeployed without affecting the wider application. The cycle of system-wide upgrades, testing, and lockdowns becomes a thing of the past.

They're business focused – Each microservice is devoted to a specific business capability and priority. Compared to monolithic software, in which teams focus on platform-wide technical requirements, such as server-side logic, UI, databases, and technology layers, the microservices model allows code changes to target a focused business value. Each microservice is owned and maintained by a dedicated team. If a team needs to make a specific UI change, it won't impact the work of other microservices or teams.

They reduce risk – With traditional systems, providing an update or configuration change means applying it to the entire application and the data structure. This triggers code freezes to accommodate peak, costly downtime, and potential surprises as the technology comes back online.

A cloud-native model allows productive commerce environments to remain operational during the process of updating and applying changes — without downtime, code freezes, or lost opportunities.

They're resilient – Microservices are built to cope with failure. If one microservice malfunctions, it can be switched off and repaired without impacting the other applications.

They're diverse – Microservices allow programmers to use a variety of technologies and platforms, enabling them to be as creative as possible when solving problems. Netflix, for example, which is responsible for 30 percent of global web traffic, encourages its developers to save time by using code libraries established by others, while also giving them the freedom to flirt with alternative solutions when needed.

They're evolving – Microservices are infinitely expandable, meaning you can attach new components when the need arises, ensuring the constant evolution of your applications. When you're on the cutting edge, you can maintain a competitive advantage.

BIG BENEFITS FROM MICROSERVICES

These are just some of the characteristics of Manhattan's microservices architecture that enable retail customers to be more agile and more responsive to business opportunities.

Discover how cloud-native computing can help your organization.

Call +**1 (877) 596-9208**, email **info@manh.com**, or visit **manh.com** for more information.

LEARN MORE

Manhattan Associates®