

Top Five Ways Microservices Are Changing Retail

Microservices have arrived, and they are quickly gaining adoption.

IDC predicts that [90% of applications will utilize microservices architecture](#) by 2020, with many of the world's largest brands, including [Amazon](#), [Google](#), [Netflix](#), and [Uber](#), moving to a microservices architecture approach.

The intense competitive pressures experienced by most companies will continue to drive the adoption of microservices architecture in software development.



Why Are So Many Enterprises Moving to Microservices?

Here Are the Five Main Reasons:

1. Future-Proofing

A large monolithic-designed application is not highly adaptable to changes, particularly unanticipated changes. And in this age of lightning-quick evolutionary leaps in both technologies and business processes, the ability to readily adapt to any changes coming around the corner is crucial. As a collection of small, independent processes, a microservices-designed application is far more suited to adapt to changes in the business environment. As a recent [Forbes article noted](#), microservices can enable you to "...profit from disruption rather than succumb to it."

2. Scalability

The need to scale an application often becomes a good news / bad news scenario. The fact that a boom in business necessitates scaling is the good news. The difficulty that will be encountered in scaling a monolithic-designed application is the bad news. When your application is built as one integral whole, there is no choice but to scale the entire application when any one part of it needs to scale. But when the application is comprised of many modules, each module can be scaled independently, allowing for efficient use of resources and more granular control of operations.

3. Digital Transformation

Nearly [two-thirds of retailers are implementing digital transformation](#) initiatives, or are on the verge of doing so. Microservices will play a significant role in the process by providing the agility that is essential to enabling digital transformation.

4. Flexibility

Need one of your application's business modules to be in Java? Will C++ be best for another? With a microservices architecture, each module of an application is a standalone unit, and can be designed using whatever parameters are best for that single module. This also enables quick adoption of new technologies and allows you to tap into different talent pools to combine the "best of the best" at what they do.

5. Time to Market

In today's ultra-competitive retail environment, the time required to move an application from concept to market is crucial. The market is rarely kind to latecomers. By its very nature, a microservices-based application lends itself to the simultaneous, parallel development of multiple modules, a process that helps trim time to market to an absolute minimum. It also means that you can quickly deploy updates to applications (CI/CD – continuous integration and continuous delivery), as it's much easier and safer to update little parts of an application than to update one big monolithic application.



A Microservices-Enabling Database

All of the above advantages are great, but you won't be able to truly have them unless you have the right kind of database—that is, a database that enables a microservices architecture.

The traditional relational database architectures long used by monolithic applications are not likely to be suitable for use in a microservices-architected application, and that's because relying on traditional RDBMS concepts such as ACID capabilities, referential integrity, and unique indexes can work inside one microservice but typically break down across distributed microservices with individual data stores.

Microservices-based applications typically [require multiple data models](#). A multi-model database is often the best choice for a microservices application because it allows the data model flexibility required to build a real microservices-based architecture.

The Bottom Line

If there's one thing we can conclude from all of the above, it's that developers are more empowered than ever. Via microservices, they have the nimbleness and flexibility they have always asked for, and this is allowing them to build more powerful applications than ever before.

However, they also need the right tools for the trade.

Relying on an RDBMS as your only data store in a microservices-based architecture is short-sighted; being able to bring a NoSQL data store into the mix when you need consistently low latencies in a linearly scalable, extremely highly available environment is important to enterprises who want to do microservices well.

However, within the multi-model/NoSQL world, there are many different types of databases serving many different needs.

DataStax's active everywhere database is uniquely positioned to serve a microservices-based architecture.

About DataStax

DataStax delivers the only active everywhere hybrid cloud database built on Apache Cassandra™: DataStax Enterprise. The foundation for contextual, always-on, real-time, distributed applications at scale, DataStax Enterprise makes it easy for enterprises to seamlessly build and deploy modern applications in hybrid cloud. DataStax also offers DataStax Managed Services, a fully managed, white-glove service with guaranteed uptime, end-to-end security, and 24x7x365 lights-out management provided by experts at handling enterprise applications at cloud scale, and DataStax Distribution of Apache Cassandra, a production-certified, 100% open source compatible distribution of Cassandra with expert support. For more information, visit www.DataStax.com and follow us on Twitter @DataStax

For more information on how DataStax helps retailers with their data transformations visit www.datastax.com/retail.

© 2019 DataStax, All Rights Reserved. DataStax, Titan, and TitanDB are registered trademarks of DataStax, Inc. and its subsidiaries in the United States and/or other countries.

Apache Cassandra, Apache, Tomcat, Lucene, Solr, Hadoop, Spark, TinkerPop, and Cassandra are trademarks of the Apache Software Foundation or its subsidiaries in Canada, the United States, and/or other countries.

June 2019