



PRODUCT BUILD | APPLICATION MODERNIZATION

# Grocery Retailer

Retail Sector

Our client is a popular value supermarket chain with stores in over 30 countries worldwide. Their operational and customer data had grown to a point where it had become difficult to manage effectively with the incumbent database solution, CosmosDB. The client selected MongoDB as the database solution to mitigate their issues, recognizing it as technically more efficient and cost-effective.

As gravity9 had previously completed projects with the supermarket chain and MongoDB we were approached to enable a database transition from CosmosDB to MongoDB. Three key areas were addressed: gravity9 adapted existing client technologies to be compatible with both CosmosDB and MongoDB, allowing for a staggered transition from one to the other, avoiding

the risk of a “big bang” all-at-once approach. C#.NET reusable components were developed for this project enable the client to use them as accelerators on future projects. Lastly, gravity9 provided consultative architectural advice on managing the migration of billions of data records, leveraging a Kafka-based pipeline to move and process data in near real-time.

gravity9 completed the project, providing our client with an efficient yet secure transition from CosmosDB to MongoDB across multiple countries, reusable software assets, and an architectural plan for their data migration. As a result, the client enjoys improved efficiency and reduced costs in this area of its data systems.



## Review of Challenges

Our client, a leading international value supermarket chain with over 13,000 stores in over 30 countries globally and has been trading for over 90 years. As a result, it has amassed huge volumes of operational and customer data, with the data for just one country's operations often equaling more than a billion individual documents.

Managing this data and database (specifically within their receipts management sector) had become cumbersome and costly. Our client sought to modernize with a flexible solution that could scale with future growth, respond rapidly to data requests, and reduce data storage costs.

Our client identified that NoSQL database provider MongoDB could best serve their needs and approached MongoDB for the project. As gravity9 had successfully completed previous solutions for both our client and MongoDB, we were engaged in delivering this migration.

“... (the client) sought to modernize with a flexible solution that could scale with future growth.”



## Utilized Technology Stack

- **Cloud:** Microsoft Azure
- **Database:** MongoDB Atlas, Cosmos DB
- **Backend:** .Net, C#, Kafka, Kubernetes

## Our Solution

At the project's commencement, the client employed a .NET codebase with a Microsoft CosmosDB data layer, which they required for operations across their business.

To ensure minimal downtime during the data migration, gravity9 leveraged our considerable experience from previous MongoDB and .NET projects to provide not only the development of the service, and a live data migration to MongoDB but also a long term migration plan for mitigating all risks relating to the transition from Azure CosmosDB to MongoDB.

Reusable C#/.NET components were used across multiple client .NET services, offering a cohesive, uniform solution that could be deployed several times while retaining best practices without the need to re-develop individual components each time across various microservices.

Although implementing a Kafka-based flow was not part of this project's scope, gravity9 consulted with the client to provide architectural advice on a successful Kafka flow and assets to assist in future architectural development.

## Our Approach

gravity9 conducted thorough requirement gathering to understand the scope of the project, its challenges, and priorities for the client. Our client was keen to ensure a smooth transition from CosmosDB to MongoDB, and it was agreed that a "big bang" approach – while rapid – would introduce undesirable risks to the transition.

Initially, the client's .NET codebase was set up for CosmosDB data access layers, so gravity9's solution was to adapt this codebase to manage MongoDB simultaneously.

Combined with a Kubernetes dynamic configuration, this enabled a gradual deployment country-by-country, minimizing risk and providing an easy rollback in case of any issues.

As several .NET services needed attention during this project, reusable data access libraries in C#/.NET were built to manipulate data, manage MongoDB connectivity, and process data transactions. Azure Artifacts and NuGet packages were leveraged to allow these reusable components to be applied easily, using best practices, across all services. These components can be used across other .NET services in the future, providing valuable functionality to the client for areas outside of the project remit (receipts management).

The project was to ensure a successful transition from CosmosDB to MongoDB rather than to complete the data migration of the client's billions of records.

gravity9 also provided architectural advice on how best to leverage Kafka streaming and incremental load capabilities with existing batch load tools to build a resilient solution for dealing with the data migration.

## Subsequent Outcomes

gravity9 delivered a successful database transition that allowed existing systems to communicate effectively with MongoDB. For our client, this provides, this provides significant benefits in the form of cost savings by maintaining large volumes of data while simultaneously achieving faster response times despite high numbers of data requests.

While not directly aware of the project, the supermarket's customers' will benefit from MongoDB Atlas' accurate, robust data stores when interacting with the business.

The completion of the project also marks a significant milestone in part of a larger client project to replace the majority of their non-relational data storage with MongoDB. Our client is empowered with reusable .NET components delivered by gravity9, which can be employed in other business areas. They also benefit from architectural strategies to ensure a smooth transition of records using a Kafka-based pipeline to assist in live data migration.

**Visit our Insights page for more articles about emerging technology trends, the IT industry, interviews, and more!**

---

gravity9 leveraged our considerable experience from previous MongoDB and .NET projects

---



---

“... (our client) is empowered with reusable .NET components delivered by gravity9.”

---

Visit our Insights page for more articles about emerging technology trends, the IT industry, interviews, and more!