



CASE STUDY

Improving Collaboration with the DataKitchen DataOps Platform

THE PROBLEM

A large fortune 500 healthcare company struggled with a complex data landscape spread across numerous locations. It was challenging to collaborate and work across teams and data centers.

One team located in New Jersey operated an on-prem data warehouse. They shared data with another independent analytics team located in California. This team used Azure and an Azure toolchain.

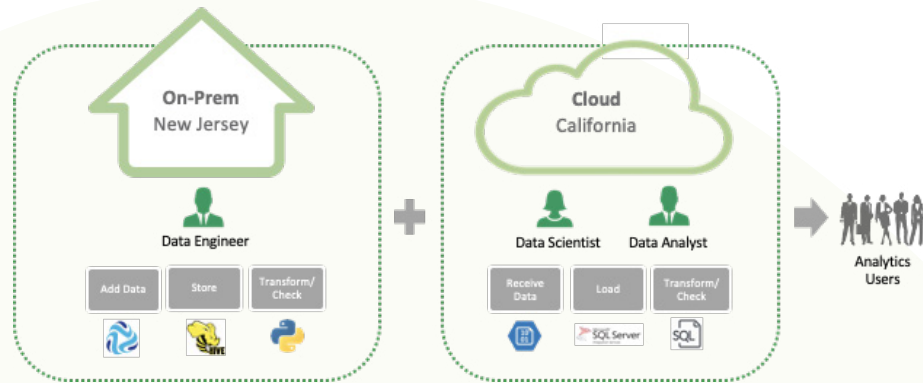
These teams were not coordinated so this seemingly simple transfer of data resulted in numerous problems. Any time the New Jersey team made a change to their schema, they hoped that it would not break anything further downstream. The California team had no visibility when a change was made. Each time the two schemas diverged, it led to errors and delays. Trust was lost and finger-pointing followed.

THE SOLUTION

Using the DataKitchen platform, each team built a DataKitchen Recipe (pipeline) that orchestrated their local workflows and toolchains in their respective environments. The Recipes included automated tests to check the tables before data transfer. For example, the on-prem team's Recipe moved data from a source to a data store, ran a transformation, and performed quality checks on that data with Python.

The Azure team's Recipe received the data, moved it into storage, loaded it into an SQL server, transformed the data, and tested it with SQL. These Recipes and tests are owned and maintained by the respective teams (figure 1).

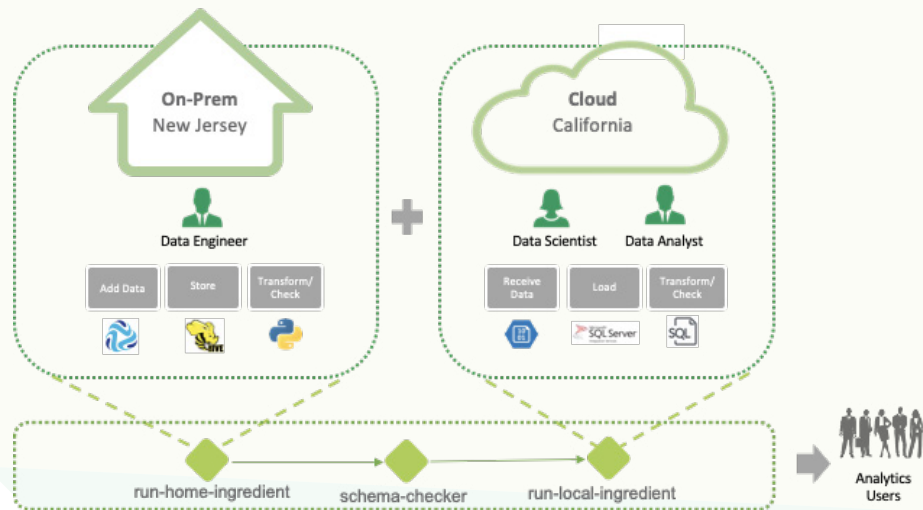
FIGURE 1:
Each team created a Recipe using their respective toolchains and environments.



In addition to the tests in the individual Recipes, the teams added a test automation node that reads the full schemas from both tables and compares them every 10 minutes. If the schemas do not match, then the test fails and sends an alert to the appropriate users so they can correct the drift before the next data transfer.

Each team’s Recipes were saved as an Ingredient in the DataKitchen Platform. An agent-based architecture enables access to both the on-prem and the Azure cloud systems. As a result, the Ingredients could be meta-orchestrated into one workflow by a centralized Recipe that calls the local Recipes, enabling seamless collaboration across locations, environments, tools, and teams. (figure 2)

FIGURE 2:
Each team’s Recipes were orchestrated into one seamless workflow.



RESULTS

The common DataOps Platform has made it significantly easier for teams to collaborate effectively while continuing to work in different locations and data centers and using their respective toolchains. Automated testing was also critical to increase the quality of the analytics. The teams now receive a warning if the data or schemas

drift and can make any necessary updates before errors reach end-user customers. With this new process in place, the teams are significantly more productive. They spend much less time finding and fixing surprise errors, while the level of trust between teams has increased dramatically.