### **Vizient Azure Cloud Migration**

**Business Problem:**

Vizient's legacy on-premise systems, which depended heavily on third-party applications and in-house servers, were increasingly cumbersome and not cost-effective. The existing "lift and shift" approach posed several operational challenges and the on-prem system's outdated security model and reliance on file storage were clear roadblocks to efficiency and scalability.

**Solution:**

* **Application Background:** This migration included 3 core systems each hosting around 15-20 services linked with internal web applications. These web applications utilized a mixture of software technologies - React, Angular, .NET, and more.
* **Phased Approach:** An agile process was adopted. In Phase 1, the systems were migrated to cloud with minimal changes and in Phase 2, the team minimized 3rd party dependencies and aligned the code.
* **Upgraded Security:** Before migration, Vizient's security model was revamped, strengthening its defenses against potential threats. Modern security measures such as SSO were incorporated into the system architecture.
* **Infrastructure Evolution:** A framework, which took four years to develop, was established, standardizing Azure service utilization, security protocols, and cost-efficiency strategies. This blueprint expedited migrations significantly.
* **Optimized Storage:** Instead of traditional file storage, Blob storage was used from the onset. This decision not only optimized server configurations but also brought down associated costs.
* **Data Processing:** The backend's data processing layer was transformed, shifting from traditional SQL databases to augmenting Azure Data Factory (ADF) and Databricks. This change was paired with SQL schema updates and new data migration scripts. Introduction of ADF and Databricks minimized the workload related to data migration on the webservers in the cloud and enabled operational efficiency.

**Team & Timeline:**

Over 1.5 years, a 10-member team of software consultants undertook this mammoth task. The entire web application was redesigned with front-end changes in Angular and necessary updates in SQL. In addition, the introduction of ADF and Databricks required substantial code additions on the backend.

**Business Impact:**

* **Cost Efficiency:** The deployment of a standardization framework, which took 1.5 years to establish, resulted in significant cost savings. Utilizing Blob storage reduced server costs, while the adoption of ADF and Databricks offered a more cost-effective model for data migration and processing.
* **Enhanced Security:** The revised security protocols ensured the safeguarding of data during and after the migration process.
* **Data Management:** By introducing intermediate layers like ADF and Databricks, data transfer to cloud servers was managed more effectively, ensuring data integrity.

**Challenges:**

* **Cost Management:** Monitoring was vital to avoid unanticipated costs, especially during the development phase in the Azure environment.
* **Data Migration:** Moving vast volumes of data, especially with schema changes, required meticulous planning and validation.
* **Testing:** The extensive data-driven nature of the applications demanded rigorous testing to ensure all data was migrated correctly.

**Results:**

Vizient now boasts a streamlined, cost-effective cloud infrastructure on Azure, augmented with modern security standards and optimized data processing capabilities.

**Lessons Learned:**

* **Plan in Phases:** Adopting a phased approach can smooth out complexities, especially in large-scale migrations.
* **Standardization is Key:** Investing time in developing a standardization framework can accelerate migration and reduce associated risks.
* **Continuous Monitoring:** Especially during migration, it's crucial to monitor cost and performance metrics to prevent potential pitfalls.