



Redesign & Implementation Of Technical Architecture

(Premium Beauty & Wellness Brand)


Identified and implemented appropriate changes to the client's technology stack and data systems to simplify data flow and enable **robust data management and reporting capabilities**

REDESIGN AND IMPLEMENTATION OF TECHNICAL ARCHITECTURE FOR PE-OWNED COSMETICS COMPANY


ABOUT THE CLIENT

Client is a **premium beauty & wellness brand** in North America


SITUATION

- 
- Client operational functions involved multiple tools and technologies, with convoluted workflows, leading to **undue complexity and inefficiencies in data processing**
 - Merilytics partnered with the client to assess, identify and **recommend appropriate technology stack** to simplify the data flows and infrastructure, and also enable more robust data management and reporting capabilities

VALUE ADDITION

- 
- Developed a deep understanding of the client data systems - CRM, ERP and other in-house applications, and technical architecture**, to audit the operational data flows
 - Conducted a **comprehensive study of various data engineering tools** in the market, **presented a comparison matrix** across several parameters including **technical features, flexibility of use, scalability, source code version control and pricing**
 - Developed POC demonstration of top-3 tools** to help client Technology team to make an informed decision
 - Deployed the new technology stack** for efficient data engineering in client environment
 - Designed and implemented final dashboard reports** with the **new architecture**, and **enhanced** monitoring and tracking capabilities

IMPACT

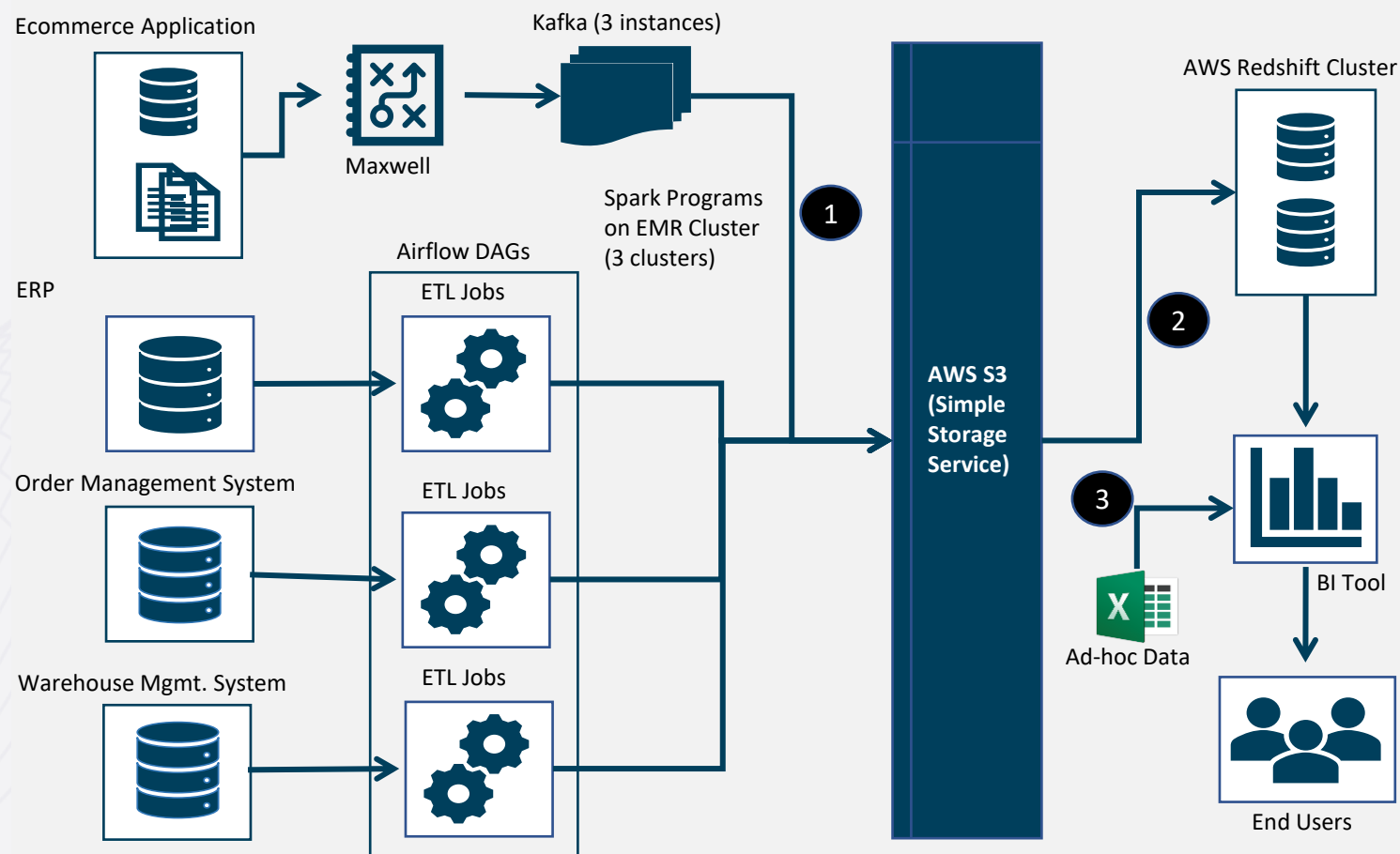
- 
- Enabled client to move to a **simpler technology stack**, but with more **reliable data flow and reporting infrastructure**
 - Enhanced monitoring capabilities and improved accuracy of reports, drove **operational efficiency and improved decision-making process significantly**

REDESIGN AND IMPLEMENTATION OF TECHNICAL ARCHITECTURE

Feature	Before	After
Tools and Technologies	Multiple tools were used for data management leading to a complex and unstable architecture	<ul style="list-style-type: none">• End-to-end ETL tool (Talend Data Management System) is deployed, with ready to use connectors for multiple data sources, destinations and supports various formats (CSV, XML, JSON, RDBMS and NoSQL databases)• Single application to connect, transform and load volumes of data
Ease of Use	Code was written in Python/Java for the ETL process which was time consuming and prone to errors	<ul style="list-style-type: none">• In-built capability of ETL job functions, which is workflow based (“drag and drop”) and less error prone• Workflow can be configured easily and scheduled to run at pre-defined intervals
Source Version Control	ETL job code was separately stored in a version control system and synchronization between the various environments (dev, pre-prod and prod) is managed manually	<ul style="list-style-type: none">• Ready configuration setting to any version control system• Synchronization between various environments is self-managed without any manual intervention
Costs	Apart from the data sources, virtual machines had been dedicated to run Maxwell, Kafka (3 clusters) and Spark Jobs on EMR cluster (3 clusters), leading to a cost of ~\$50k per year	Single high-end virtual machine replaced several hardware components, thus reducing the cost by 50%
Monitoring capabilities	<ul style="list-style-type: none">• Monitoring of jobs was done on a separate application (Airflow) where all the ETL jobs were configured.• Tracking was made possible by pushing Slack notifications to the end users	Monitoring and tracking capabilities were in-built into the ETL tool, and can interact with several notification applications including Slack

TECHNICAL ARCHITECTURE (BEFORE)

Technical Architecture (Before)

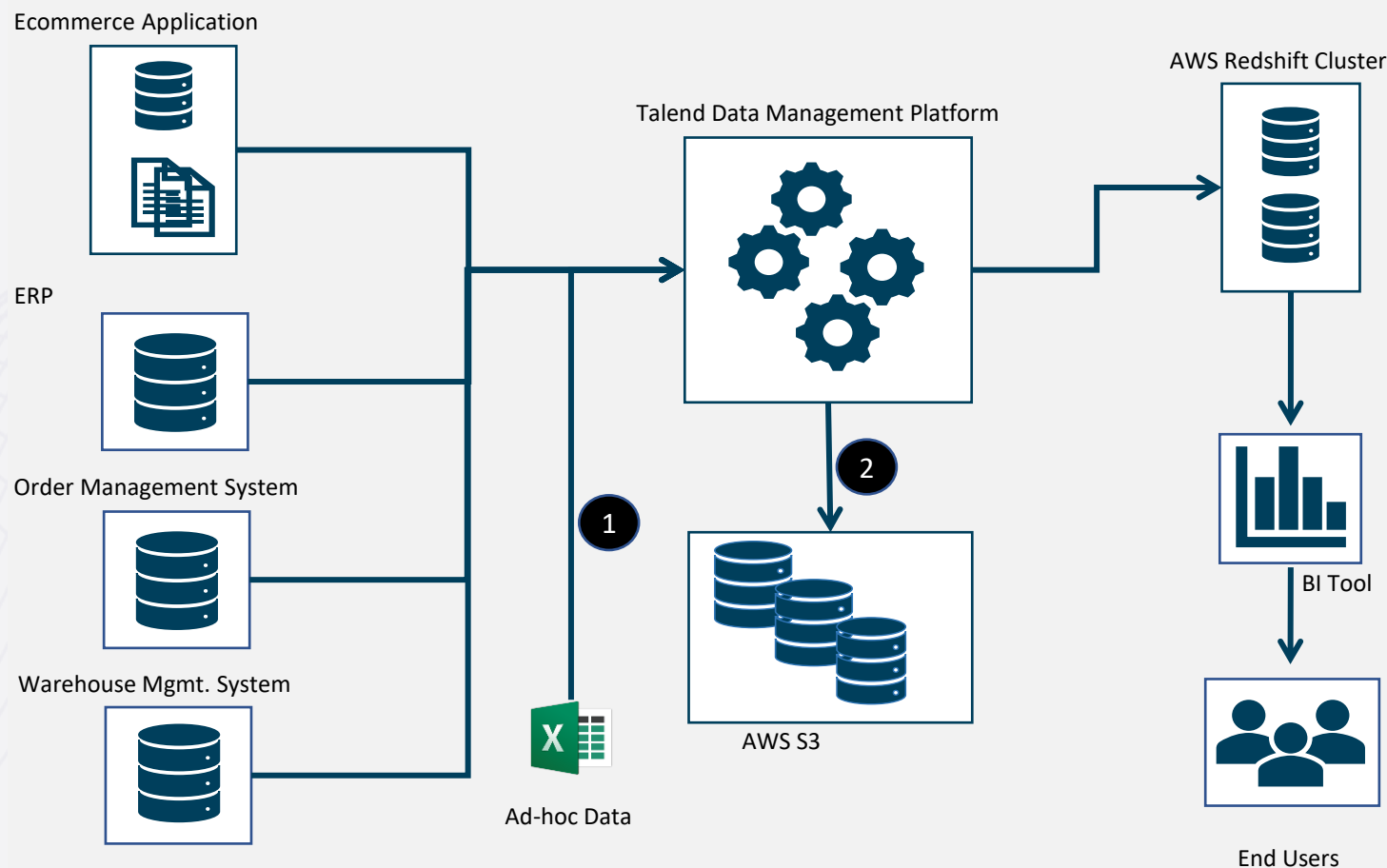


- All ETL jobs were coded in **Python/UNIX shell scripts** and they are managed in Airflow, a tool for describing, executing, and monitoring workflows
- Due to the **complex nature of ETL jobs** and usage of too many technology components, monitoring and tracking is time consuming and error prone.
- **All transformation is done in Redshift cluster (configured for dense storage)** using programs written in Redshift, which is inefficient and time consuming
- The **cost of maintaining the infrastructure was high** due to dedicated VMs, MapReduce and Redshift Clusters
- **Many users with Editor privileges** caused far too many datasets and dashboards leading to ineffective monitoring & tracking of datasets

- 1 Data from Kafka (3 instances) flows to S3 bucket through Spark programs which run on EMR cluster (3 clusters)
- 2 Data from S3 is moved to Redshift cluster through COPY command
- 3 Ad-hoc data from Excel sheets are loaded and used in conjunction with existing datasets to create their own reports. Later, the combined datasets are published and used by other users leading to several untraceable dashboard reports

TECHNICAL ARCHITECTURE (AFTER)

Technical Architecture (After)



- All ETL jobs were converted into **workflows** in Talend using the “**drag and drop**” pre-defined objects, making it easier to visualize the jobs
- All jobs could be **viewed and monitored in a single screen**
- **Source code for all the jobs** is generated automatically and can be configured to any version control system (GitHub, BitBucket etc.)
- Provision to write and execute **Python/Java** code as part of the workflows and supports complex transformation of data
- The **TCO of new solution is much lower than** existing infrastructure
- Process changes ensured that **all input data is routed through the tool**, ensuring only approved datasets are created and users do not have editor privileges

1 Editor privileges for users was revoked and all input data was routed through Talend job

2 Data save in S3 and Redshift was created as parallel flow instead of sequential to reduce time for data loads