Standardization of System Integrated Data Engineering Architecture

Purvesh Jadhav
Data Architect
Slalom Inc, 255 S King St
#1800, Seattle, WA 98104, US

ABSTRACT
Data Engineering plays an important role in the Data Science and Analytics industry to ensure that the data availability is seamless, efficient and accurate for further analysis. Rapid evolution of data engineering facilitates the need for robust and standardized approaches of system integration in data engineering. With increase demands for scalability, reliability, efficiency and inconsistent methodologies can lead to fragmented solutions, operational inefficiencies and high maintenance cost, thus standardization of system integrated data engineering solution helps to mitigate these challenges achieve the timely and effective data consumption, storage and availability. Also, managing these system integration solutions becomes challenging if business runs under different industries, and different platforms. Hence it is very important for data engineering architects to standardize the system integrated solutions when it comes to data engineering project implementation. This article explains feasibility of standardizing the system integrated data engineering architecture and proposing a framework that emphasizes uniform practices in areas such as data ingestion, transformation, storage, and retrieval across diverse platforms by conducting a case study of leading Global Online Retail Company.

Keywords
AWS (Amazon Web Services), System Integration, ETL pipelines, Cloud Native Solution, Data Engineering.

1. INTRODUCTION
Data Engineering is a channel to bring together all required data entities and provides groundwork for analysts and data scientists to identify patterns and make decisions which can eventually help business increase profitability and reduce costs. This concept is widely used in industry across multiple business verticals and working in dynamic environments. In the rapidly expanding field of Data Engineering, system integration with complex systems has become cornerstone for achieving seamless data workflows, interoperability, and efficient decision-making. As organizations increasingly rely on diverse platforms and tools to process vast amounts of data, the absence of standardization in implementation leads to operational inefficiencies, inconsistencies, and difficulties in scaling systems. Typical Data Engineering Model has been shown below to understand entities involved.

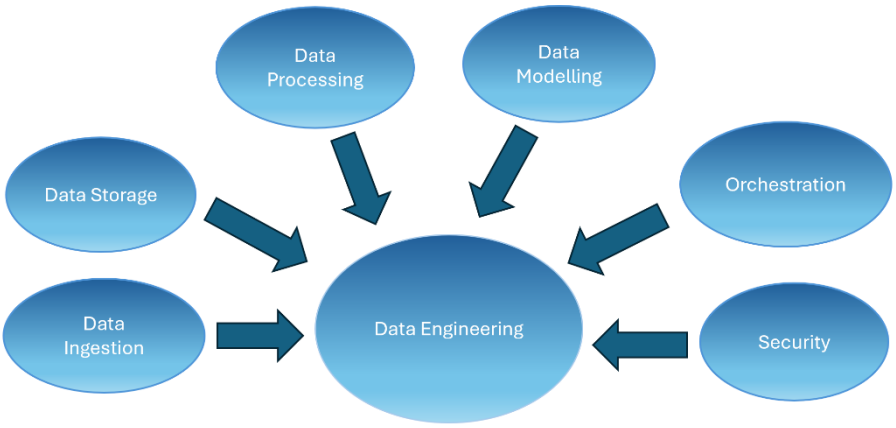


Fig 1: A Typical Data Engineering Model

Periodically based on SLA (Service Level Agreement) provided.

Detail breakdown of data Engineering Model helps to identify entities involved. “Data Ingestion” is generally the first step where data is collected from multiple heterogenous sources such as API’s, databases, file systems or streaming services. “Data storage” is the next step where collected data is stored in raw format in secure storage location such as database, AWS S3 bucket etc. “Data Processing” is where data is cleaned and processed according to business rules and mapping requirements. To be able to successfully create mapping document data modeler needs to be involved to study data to eventually create “Data Model” to achieve business objective. “Orchestration” is a step where data engineering Architect put emphasis on scaling and scheduling the entire flow of fetching data to make it available for data analyst. Data generally includes customers or users PII (Personally Identifiable Information) information, that makes “Security” step more crucial, and industries wants to make efforts to make this data

As more data sources and more business problems are involved in data engineering, the architecture becomes more and more complex to manage. The core and integral part of data engineering is to ensure ingestion and storage of data

stored at secure location with limited access authorized individuals.

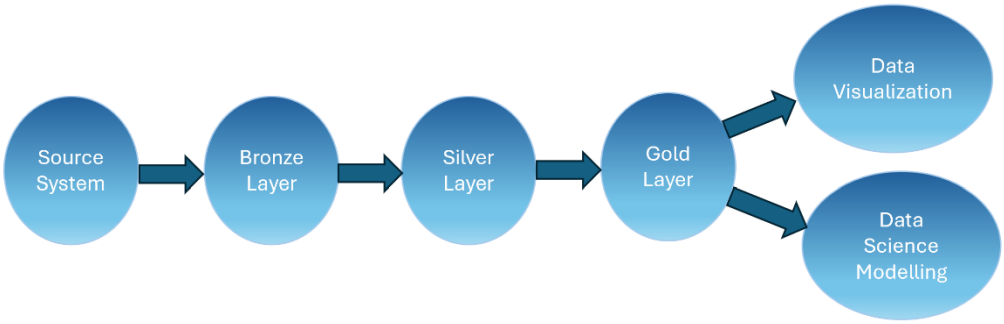


Fig 2: A Typical Data Engineering Workflow

Managing these Data Engineering model entities and workflows can be challenging for the Data Engineering team, particularly when ensuring a highly available, reliable, and scalable architecture across diverse source systems. Therefore, standardizing the implementation of an integrated Data Engineering solution is crucial to keeping costs and efforts manageable, ultimately benefiting industries in the long term.

2. LITERATURE REVIEW

Managing, scaling, and optimizing data engineering solutions across diverse industries, sources, and use cases becomes unfeasible without standardizing the approach. Standardization enables seamless onboarding of new sources quickly and efficiently, minimizing overhead costs and efforts for the Data Engineering team, especially when dealing with highly unique and complex requirements.

Sultan Yerbulatov [1] has insisted on the importance of efficient processing and analysis to make informed decisions especially for large organizations. Also articulated that big data integration is complex process and focuses on the development and maintenance of data architectures that ensure the efficient flow of data from its sources to end users benefits greatly to organizations.

Professor Marco Iansiti (Harvard Business School) [2] asserts that standardization of data governance and interoperability of systems enables organizations to adapt to privacy regulations like General Data Protection Regulation (GDPR).

The Data Engineering Model entities implementation results into Data Engineering Workflow which is facilitated by data architect and data engineers to make data available for Data analyst for Visualization and Data Scientist to generate a model as per business objectives. A typical Data Engineering workflow is shown below. Data Engineers identify the source system and secure connection mechanism to fetch the data and store it into bronze layer in raw format. The minimal data processing has been performed to standardize and easy to access data in readable format results into silver layer. Gold layer is where most of complex transformation and business logic gets implemented to connect multiple data sources resulting into data model. The PII data in gold layer is generally hashed out and non-hashed data access provided to authorized users only.

Professor V.J. Reddi (Harvard University) [3] raised concerns over scalability issues in creating and maintaining machine learning datasets, the lack of standardized frameworks for automatic dataset generation. He emphasizes the importance of standardized tools and collaboration in building resilient and scalable data systems in the Data Engineering world.

Professor Michael Stonebraker (MIT) [4] a pioneer in databases and big data, frequently addressed the "one-size-fits-all" problem in data architecture, where he provided importance of generic solution in data engineering.

Industry experts believe the following issues are prevalent: over-engineering of ETL pipelines, resulting in high maintenance costs; lack of modularity in architecture, making future changes expensive and complex; and poor observability and monitoring of data workflows, leading to undetected issues. These challenges can be mitigated by standardizing data engineering solutions.

Many professors and researchers have emphasized the importance of standardizing Data Engineering architectures. However, no significant research or articles have been published advocating for a standardized, system-integrated solution in Data Engineering. This article aims to address that gap, achieving key milestones in proposing a standardized approach to solving common challenges in Data Engineering implementations.

3. PROBLEMS TO SOLVE

Implementing a standardized Data Engineering solution enables organizations to maintain and scale their systems effectively, ultimately reducing costs and alleviating implementation burdens in the long run.

When designing a data integrated solution, the following common problem statements need to be solved.

1. Mode of communication with external systems.
2. Make the choice between On-Premises or Cloud Native solution.
3. Secure storage of credentials while fetching data from external systems.
4. Process to fetch required volume of data based on schedules.
5. Storage of raw data at secure location and avoid overhead cost of storage to backfill.
6. Methods to handle PII and GDPR requirements.
7. Methods to implement logging mechanisms.
8. Decide Orchestration solution to schedule required tasks.
9. Storage of raw data at secure location and avoid overhead cost of storage to backfill.

Below is the diagram which provides visual representation of common problems to be solved in data engineering while

designing the solution.

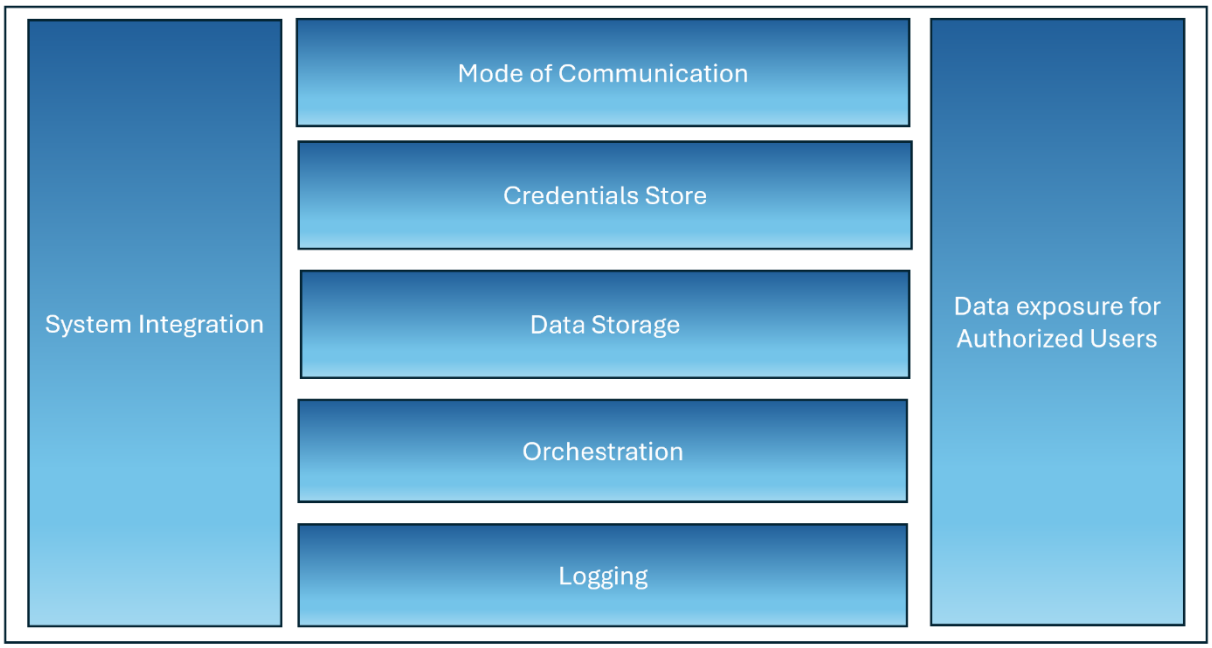


Fig 3: Primary Problems to solve in Data Engineering

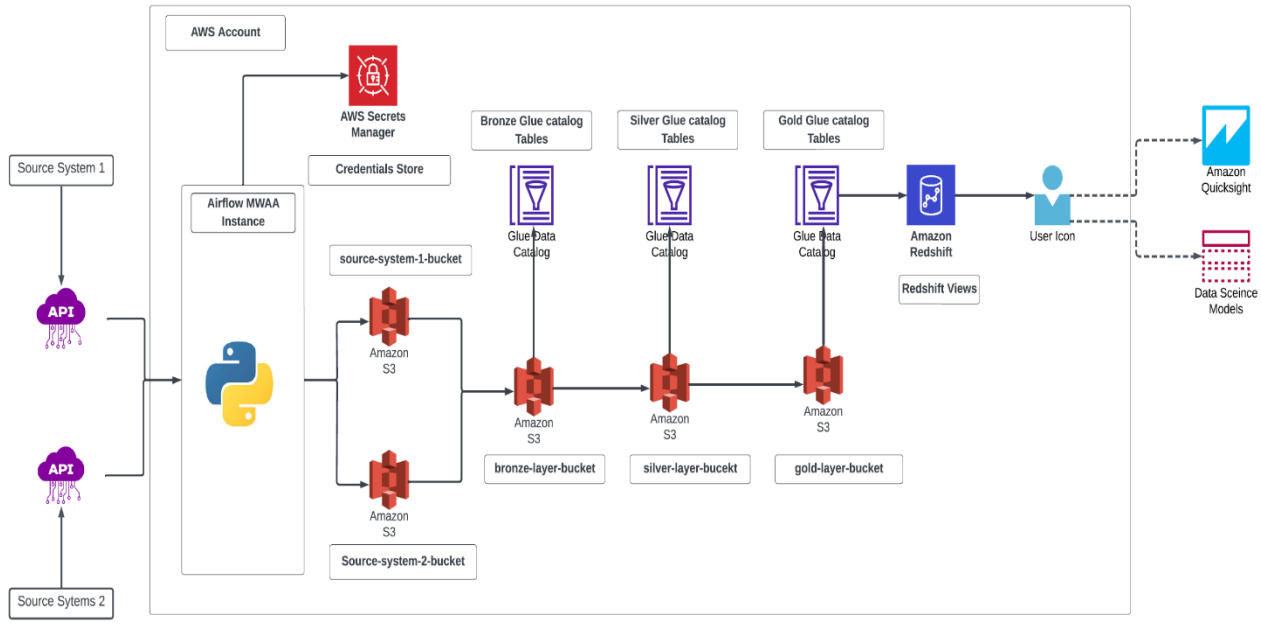
4. A CASE STUDY ANALYSIS

The case study has been conducted in a leading global online retail company to understand the system integrated data engineering solution that are offered in data engineering.

The key pain points were identified by studying and observing existing Data Engineering implementations. The company wanted to use APIs to fetch data from external survey systems to analyze customer sentiments. However, the company already had another Data Engineering solution designed to address the same problem, but it lacked standardization, making it insufficient to handle additional workloads and logging mechanisms. In this solution, AWS cloud services [5] such as EC2 instances, S3 buckets, and Redshift were utilized, which created challenges in maintaining and scaling the solution in the future.

To address these challenges, subject matter experts in Data Engineering collaborated to identify the key factors impacting the standardization of system-integrated Data Engineering solutions. After conducting extensive research, they proposed solutions to address each problem. Below are the outcomes of their research:

1. API for Data Communication: The company chose to continue using APIs as the mode of communication. APIs facilitate faster data retrieval and are widely accepted across the industry.
2. Cloud Adoption for Cost Efficiency: The company adopted cloud native solutions to reduce infrastructure maintenance costs and resource overheads. AWS managed services were selected for their high reliability, scalability, and minimal downtime, aligning with industry standards.
3. Secure API Access: To ensure secure API access, credentials are stored in AWS Secrets Manager, a secure and commonly recommended practice for managing sensitive information.
4. Raw Data Storage: Raw data is stored in a secure S3 bucket with restricted access. The data, stored in JSON format, allows for backfilling when needed.
5. Glue Tables for Raw Data Management: AWS Glue tables were used to store raw JSON data as a column, alongside the S3 file path and insertion timestamp. This setup enables Data Engineers to query data from Athena for troubleshooting.
6. Silver layer Data Transformation and Standardization: Minimal transformations were applied to flatten the JSON data and store it in a columnar format. The data from multiple systems was standardized using Glue Catalog, simplifying JSON field access and enabling quick anomaly detection.
7. Gold Layer Mapping and Data Transformation: Experts utilized a data modeler to create a mapping document for the gold layer, incorporating business logic and required transformations. The gold-layer Glue tables were designed for use by Data Analysts and Data Scientists.
8. PII and GDPR Compliance: Separate Glue catalog tables were created to store Personally Identifiable Information (PII) and GDPR data. The PII tables used primary and foreign key relationships to associate with gold-layer tables. Sensitive PII data was hashed using the SHA-256 algorithm, with the hashed and original values stored in the PII tables.



Flg 4: Standerdized Architecture for System Integrated Data Engineering

Data Exposure to External Users: Data was exposed to external users through the gold-layer tables. External views were created in Redshift to join datasets from multiple systems, enabling flexible changes to data views without affecting the source data.

9. Incremental Data Loading: Incremental data loading was identified as a suitable strategy for daily data updates. This approach fetches only new data, reducing storage and execution costs while improving job execution speed.
10. Python for Development: Python was chosen for development due to its popularity in Data Engineering and ease of adoption.
11. CI/CD for Deployment: The solution was deployed using a CI/CD approach, with AWS CDK facilitating infrastructure creation and deployment. Git was employed for version management, enabling seamless collaboration.
12. Automated Testing: Automated test cases were implemented in Python and integrated into the CI/CD pipeline. Deployments proceed only after successful test execution, ensuring robust solutions.
13. Orchestration with AWS Managed Airflow: AWS Managed Airflow [6] was selected for orchestration to avoid manual patching and provide a user-friendly interface for job execution and log management.
14. Pipeline Troubleshooting and Maintenance: Airflow simplified troubleshooting and managing multiple ETL pipelines. Its features include re-executing jobs, identifying failures, and triggering email notifications for pipeline issues.

The company had started seeing the Benefits of Using Standardized Architecture for Data Engineering.

1. Easy maintenance

- With the adoption of a standardized solution utilizing AWS tools and Airflow, maintaining data ETL pipelines becomes more straightforward. This approach enables efficient troubleshooting through detailed logs and provides clear visibility into the execution of historical schedules. As a result, the company has achieved a 45% improvement in maintenance efficiency.

2. Faster deployment

- The integration of CI/CD pipelines enables real-time deployment into production with zero downtime. As a result, the company has experienced significant improvements in deployment efficiency, reducing deployment efforts and time by 35%.

3. On-boarding new Data source:

- The standardized solution streamlines the process of integrating new data sources. With predefined templates and automated workflows, onboarding becomes faster and more efficient, reducing the time and effort required for data integration. This approach not only ensures consistency but also minimizes errors, enabling smoother and quicker expansion of the data ecosystem.

4. Automated testing

- The implementation of automated testing within the data pipeline ensures continuous validation of data integrity and system functionality. By automating test cases and incorporating them into the CI/CD pipeline, the company can quickly detect issues, reduce manual intervention, and maintain high-quality standards. This results in faster identification of bugs, improved reliability, and greater confidence in the deployment process and reduces the testing effort by as high as 60%.

While the company is reaping the benefits of using the standardized system-integrated solutions, there are some drawbacks associated with this model, including:

1. Initial Setup Complexity

- The initial configuration and implementation of infrastructure can be resource-intensive and time-consuming. Ensuring compatibility between various tools and platforms may require significant upfront effort.

2. Vendor Lock-in

- Relying heavily on specific platforms or tools, such as AWS and Airflow, could result in vendor lock-in, restricting the ability to switch to alternative solutions without significant migration costs.

Many companies are adopting initiatives to drive the standardization of data engineering solutions, recognizing it as a cornerstone of their efforts to achieve consistency, efficiency, and scalability. By implementing standardized frameworks and processes, organizations can streamline data workflows, ensure interoperability, and reduce complexities in managing diverse data systems.

5. CONCLUSION

Standardizing system-integrated Data Engineering solutions is imperative in addressing the complexities and inefficiencies of modern data workflows. This paper highlights the feasibility and benefits of adopting a standardized framework through a case study of a leading global online retail company. By leveraging AWS-managed services and implementing best practices like incremental data loading, automated testing, and CI/CD pipelines, the company achieved significant improvements in maintenance efficiency, deployment speed, and onboarding of new data sources.

While the initial setup and potential vendor lock-in pose challenges, the long-term advantages far outweigh these drawbacks. Standardized architectures ensure seamless data ingestion, transformation, storage, and retrieval while enhancing scalability, security, and compliance. This approach empowers organizations to streamline operations, minimize costs, and foster a robust data ecosystem, ultimately positioning them for success in an increasingly data-driven landscape.

6. REFERENCES

[1] Sultan Yerbulatov, “Integration of Big Data and Data Engineering in Modern Organizations”, International Journal of Scientific Engineering and Science, Volume 8, Issue 6, pp. 11-14, 2024. https://ijses.com/wp-content/uploads/2024/06/36-IJSES-V8N3.pdf

[2] Professor Marco Iansiti (Harvard University), "Data Governance, Interoperability and Standardization: Organizational Adaptation to Privacy Regulation.", No. 21-122, May 2021. (Revised November 2023.) https://www.hbs.edu/ris/Publication%20Files/21-122_77bc83c9-3aec-44ad-8bec-0c0fa181c8a8.pdf

[3] Professor V.J. Reddi, “Data Engineering for everyone” (Harvard University), “arXiv:2102.11447v1 [cs.LG] 23 Feb 2021. https://arxiv.org/abs/2102.11447

[4] Professor Michael Stonebraker (MIT), “One Size Fits All: An Idea Whose Time Has Come and Gone” https://cs.brown.edu/~ugur/fits_all.pdf

[5] AWS Cloud Services: Leading cloud provider in industry *https://aws.amazon.com/*

[6] AWS managed Airflow (MWAA): Orchestration tool developed by Apache. https://airflow.apache.org/