International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 7, Issue 1, 120-123, 2026
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V7I1P122

Original Article

Broadband Subscriber Data Integration with ETL to SaaS

Amey Deshpande
Sr. Cloud Delivery Manager, Calix Inc. McKinney, TX, USA.

Received On: 30/12/2025 ***Revised On: 31/01/2026*** ***Accepted On: 02/02/2026*** ***Published On: 04/02/2026***

Abstract - *In the Broadband Service Provider (BSP) industry, billing data can originate from various sources like customer subscriptions, payment platforms and network telemetry. The data is often imported using Comma Separated Values (CSV) file uploads from billing systems. To get the best outcome of the Software as a Service (SaaS) platform the data must be cleaned up and standardized prior to feeding to the product. The smallest error rate in billing data can lead to revenue leakage for a business. Poor data hygiene is a leading cause of billing system projects failing to meet their targets. BSPs cannot afford to have data management as a reactive measure anymore. Ensuring data quality before feeding into critical products is imperative for a successful business, however, the scope of such efforts is not always simple. Using automation for cleaning and restructuring of data with Extract, Transform and Load (ETL) tools ensures an effective and reliable methodology of data accuracy. Product adoption is as much reliant on operations and training, as it is on high accuracy in billing and network data. In modern broadband services, having healthy data is key to a strong foothold in the BSP market.*

Keywords - *ETL, Streamsets, Kafka, SaaS, CSV Data Processing, Data Standardization, Telecom Analytics.*

1. Introduction

At a high level, a BSP in the Southeast was largely focused on installing new hardware, testing internet connectivity, turning up new services, conducting field operations and regular network maintenance. In all these tasks that made up the core milestones for any service provider, there was a lot of underlying data captured across various platforms. The subscriber data was entered manually into the billing system to capture the demographic details as well as the subscription. The payment and service status data for a subscriber was also loaded in the same billing system. For monitoring and alarm notifications, the subscriber's equipment was captured in a network management system including manual entry of crucial subscriber information creating a 1:1 mapping. Finally, the very core of these operations, the access network architecture was stored as a configuration on a separate platform with daily backups and restoration. There were hundreds of thousands of subscribers, each with their unique data in the billing, network and administration systems, amounting to very large datasets for the BSP to manage. While some of the data was auto generated via telemetry and application interfaces, the majority of the data, especially the subscriber data in billing and network systems, was a manual effort. It brought inevitable human errors, misses and inaccuracies in the data over time, since the source of such data was spread throughout the BSP. It included employees across multiple teams writing to the same databases, which eventually led to a poor quality of overall data.

2. SaaS Requirements

A SaaS platform and its products are designed to provide a support, marketing and operations environment to the BSP. To adopt a product for any of these purposes, the BSP is required to bring in necessary data to integrate in either a cadence or in a real time setting. BSPs typically relied on Business Support Systems (BSS) providers and Operations Support Systems (OSS) providers to track daily billing activities along with network and field operations that go in alignment with the billing system. The SaaS platform brought together data across all platforms used by the BSP, layered with its own analytics and integration to present a seamless presentation of the whole business. The products required three sets of data:

2.1. Billing data: This data included all information related to the subscriber and was unique to their account with the BSP.

1. Subscriber name, address, phone, email, etc.
2. Service and subscription codes
3. Device identifiers
4. Other billing attributes like start date, special services, etc.

2.2. Access Network data: This data included information regarding all equipment involved in the access network from the Central Office (CO) to the last mile of the subscriber.

1. Layer-2 Ethernet and Gigabit Passive Optical Network (GPON) switches.
2. Optical Network Terminal (ONT).
3. Customer Premises Equipment (CPE).
4. Node – Shelf – Card – Port identifiers.

2.3. Layer-3 routing data: This data included information pertaining the layer-3 routed traffic for the subscribers of this BSP:

1. IP Netflows imported from the core or edge routers in the BSP's network.
2. Remote Authentication Dial-in User Service (RADIUS) identifiers for Point to Point over Ethernet (PPPoE) subscribers imported from a Radius server.
3. Internet Protocol version 4 (IPv4) subnets defined for capturing and mapping IP traffic limited to the BSP.

3. Data Convergence

3.1. Framework:

A SaaS platform, in its purpose to deliver services using the BSP's data, required an architecture to be implemented. This architecture would be designed to converge data from various to a single point in the cloud databases.

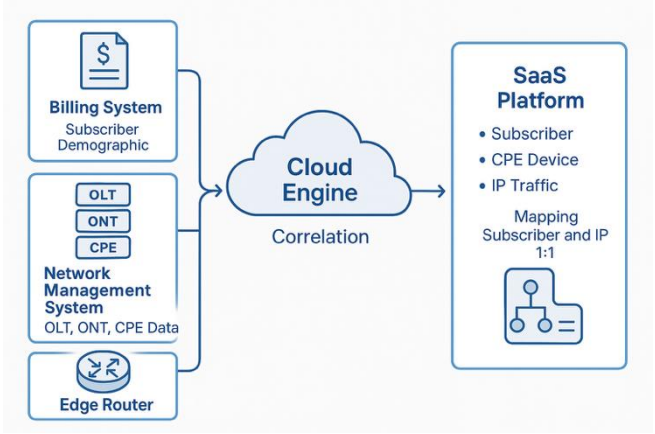

Fig 1: High Level Data Convergence to the Saas Platform

3.2. Cloud Engine:

There are various moving parts to a cloud engine especially when data extraction is involved. In today's data-driven world, organizations face the challenge of managing and harnessing vast amounts of data from various sources. This is where data lakes come into play. A data lake is a centralized repository that stores structured, semi-structured, and unstructured data in its raw format [1]. In this architecture some of the data such as telemetry from the network equipment was stored in the data lake in the raw format with minimal to no structuring. Whereas the subscriber data from the billing system had to be transformed into a pre-defined framework prior to writing to a database in cloud. Since there were millions of subscribers from various BSPs utilizing a common SaaS platform, each source of data used their own syntax and headers which resulted in requiring a structuring prior to writing in the database.

4. ETL

The BSS provider for the BSP was uploading files to the SFTP location as prescribed by the SaaS platform to be able to easily import the file. However, the structure, headers and values used by the BSS would not necessarily meet the data requirements of the product. Majority of the data needed adjustment and customized logic. To do all of that, the SaaS platform used a methodology called Extract, Transform and Load (ETL). ETL cleanses and organizes data using a set of business rules to meet specific business intelligence requirements, such as monthly reporting, but it may also handle more complicated analytics to improve back-end operations or end user experiences [2]. The SaaS platform employed ETL to:

- Extract data from billing systems.
- Reorganize the data per standardized syntax.
- Load the data into a target database.

There were multiple steps involved in the ETL process shown in Fig2.

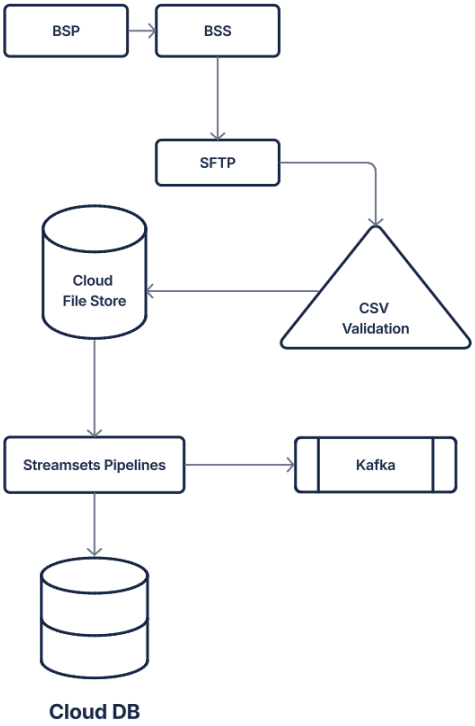

Fig 2: Steps of data ETL from BSP to Cloud.

4.1. Extract:

The first stage involved collecting the required billing data from all sources of the BSP, which in this case was the BSS provider. The most important aspect of extraction process was that it was critical to how high-quality, relevant data could be transformed and loaded into the target system [3]. The BSS provider was required to supply a flat file, or a CSV file to a pre-defined SFTP location. The file uploads would be an automated daily job on the BSS portal either configured by the BSP or outsourced to the BSS operations team for a daily upload. In cloud, the SFTP server was hosted on a Virtual Machine (VM) running Ubuntu Operating System (OS). The BSP used a unique set of username and password to login to the server. Once a file was uploaded, it would trigger a managed service for validation where it would confirm file attributes as per RFC 4180 [4]:

1. The received file is in a .CSV extension

2. No special symbols or letters other than the allowed characters and delimiters such as '-' and '.', etc. were in the file.
3. There were no white spaces in the file, which are typically found right before or after the delimiter.
4. The headers in the file did not contain any numbers or special characters of any kind.

Finally, if none of the criteria failed, the validation would pass. The files were then stored in a mounted Gluster Filesystem (FS) volume in the cloud.

4.2. Transform:

Transforming the data is the most complex of all 3 steps in ETL. Here the data ingested to the FS was read header by header, which was line 1 in the file followed by the huge datasets under these headers that formed all the comma separated values. During this stage of the lifecycle of data, extracted data are cleaned, normalized and aggregated for consistency and usability. It was important for data quality and integrity that the SaaS product take this step. Research showed that the average financial loss for an organization is $15 million a year due to poor data quality, which emphasized the need for robust ETL processes to improve data quality [3].

Streamsets was used for the ingestion and restructuring of data in this step along with Kafka. Streamsets was a widely used data integration platform crucial in helping to build, run and manage numerous data pipelines. The powerful platform was beneficial for accurately overseeing the process of batch and streaming data flows [5]. The implementation of subscriber and network data over a SaaS cloud platform touched on two complementary technologies used together for real time data processing and integration. Here's how Streamsets and Kafka worked together:

- Kafka carried out the distributed event streaming to provide high-throughput and low-latency data writes. It was the ideal choice for real-time ingestion and buffering of data streams.
- StreamSets was the data integrator that simplified the pipeline design and its monitoring. It was able to read from and write to Kafka topics using built-in connectors enabling transformation and delivery to the cloud destinations without the need of custom coding.
- In short, Kafka was used to handle data transport and durability while StreamSets managed data flow orchestration, schema evolution, and error handling.

Streamsets supported java-based regular expressions (regex) in multiple stages and functions. Regex was used for conditional routing and filtering of data based on defined pattern or logic. Few examples of streamsets regex shown below:

- It was applied to headers to iron out uppercase and lowercase alignments:

```
{
    "fromFieldExpression": "/Account",
    "toFieldExpression": "/account"
}
```

- Applied to replace the custom header to a writable header expression recognized by the database:

```
{
    "fromFieldExpression": "/'911 Zip'",
    "toFieldExpression": "/zipcode"
}
```

- Used to set a custom value to a field that was not seen in the CSV but required to be written in the database:

```
{
    "fieldToSet": "/maxdownload",
    "expression": "5000"
}
```

4.3. Load:

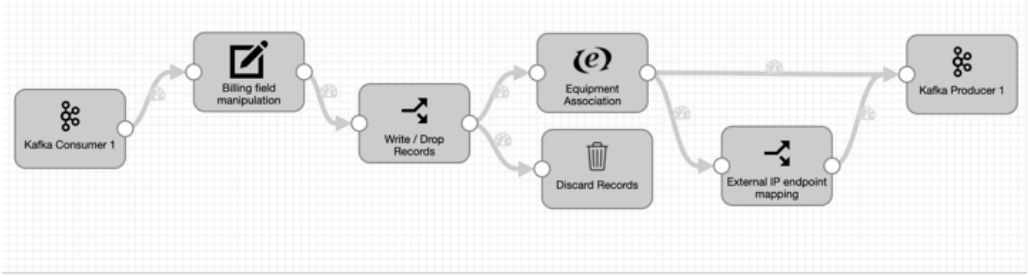

Fig 3: Streamsets Pipeline Using Kafka for data Restructuring

Based on the billing data, nodes in the pipeline were designed to bring together the billing data in a very strictly structured framework, followed by telemetry data received from the access network equipment and additional IP mapping endpoints received because of additional imports from the network management system. Using tailor made regex data structures at each node in the pipeline was eventually set up to deliver a clean and standardized dataset. The data produced in Kafka would be void of any duplicate entries and contained data headers exactly as programmed in regex. The values under the headers were either produced as is or manipulated for better alignment prior to writing. In the last stage, data Loading would move transformed data to a target system, which in this case was a data warehouse.

The cloud systems were optimized for storage and analysis, such that data was made available for business intelligence or machine learning applications. After that, data

was successfully integrated into the system in the multiple steps that made up the ETL cycle. With this structured approach, data consistency, quality and readiness for downstream analysis were ensured [3].

5. Cloud vs On-Premises ETL Deployment

Cloud ETL is managed by the cloud host, offering scalability and flexibility. On-premises ETL is managed in house, providing complete control over infrastructure and data security. It is up to the SaaS provider to architect the ETL in cloud or on-premises. The factors determining this decision are largely on the customer base, which in this case were BSP organizations, with a desire to adopt the SaaS product by integration of their subscriber data. This data had unpredictable sources among BSS providers as well as BSP in-house billing systems. Given that the revenue for BSPs is on a subscription basis, the risk is higher with long term product adoptions. With that mindset and to achieve maximum product adoption from BSPs across the country, the SaaS provider would carefully consider all the pros and cons of the ETL deployment. The below table goes over high level pre-deployment considerations for the Saas provider:

Table 1: Cloud vs on-Premises ETL Deployment

Cloud ETL	On-Premises ETL
Hosted on a third-party provider like AWS, Azure, Google.	Hosted in-house at the provider's servers and network.
Managed by cloud provider, including updates.	Managed by the provider, including all maintenance.
Highly scalable and flexible, scaled up or down on demand.	Less scalable; would need new hardware to expand, adding overhead cost.
Low initial cost - pay-as-you-go per usage. Could add cost in the long run.	High upfront investment in hardware, software, and licenses.
Less control over the underlying infrastructure; relies on the provider's security measures.	Complete control over hardware, software configuration, and security policies.
Seamless integration with other cloud services and data warehouses.	Integration requiring more effort to connect with external services.

Data integration is often the simplest type of integration to implement, but requires proper information management techniques to make the solution sustainable and cost-effective [6]. The SaaS provider, depending on the product or service their customers are adopting will architect their Data integration accordingly on-premises, or in the cloud, or a mix of both.

6. Conclusion

Small businesses typically do not have the IT depth of larger businesses and as a result they struggle to leverage their limited resources for a competitive advantage. Often, small and medium scale BSPs do not have the resources to effectively maintain their own data centers or on-premises equipment. These factors make the adoption of SaaS solutions attractive to small BSPs, as they can access high-end software without needing to install the infrastructure internally to support it. Once a solution is chosen, it is common for that solution to need to be integrated with other applications which is where the data integration and use of ETL to achieve it plays a huge role in cloud-native products [7]. The BSS originated CSV file could contain multiple duplicate rows of data per subscriber, with misses leading to empty data fields and inability of tracking certain data that the SaaS platform required for analysis. All of this was resolved by using a combination of Streamsets and Kafka to be loaded to the cloud databases as a daily job.

References

[1] Building a Data Lake: A Step-by-Step Guide with Codes and Examples, by Pradyumna Karkhane
[2] An Overview of ETL Techniques, Tools, Processes and Evaluations in Data Warehousing by Bilal Khan, Saifullah Jan, Wahab Khan and Muhammad Imran Chughtai.
[3] ETL: From Design to Deployment by Raghav, Barani, Vijay Ram, Christ University Bangalore, Bangalore, India
[4] https://datatracker.ietf.org/doc/html/rfc4180
[5] Streaming Data Ingestion into Bigquery Using Streamsets by Hareesh Kumar Rapolu, International Journal of Leading Research Publication (IJLRP), E-ISSN: 2582-8010
[6] Reliable Dynamic Data Integration Approach for SAAS Application with Their On-Premises Systems by K. Abdur Rahman Khan, Dr. P. Suryanarayana Babu
[7] Implementing and Integrating SaaS Solutions at Small Businesses by Bob Bemrose Technical Operations Support Coordinator ManageForce.