International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V5I4P110
Eureka Vision Publication | Volume 5, Issue 4, 94-104, 2024

Original Article

Cloud-Based Data Hubs and SQL Pipelines for Real-Time Financial Analytics

Karthik Allam
Big Data Infrastructure Engineer at JP Morgan &Chase, USA.

Abstract - Businesses need to get accurate, up-to-date information from more and more transactional, market, and consumer data sources in today's fast-paced financial environment. Dynamic financial analytics data solutions need to be able to alter, grow, and work quickly. Some of the old data systems that are on-site can't do these things. Cloud-based data hubs let everyone on a team and all of their tools access the same data at the same time. This makes it easier for people to get to, share, and use information. These cloud-based apps can move and change data almost instantly with SQL pipelines. This helps banks and other financial firms get rid of data silos, make decisions faster, and process information faster. When you use SQL pipelines, it's easy to automate the steps of acquiring data, cleaning it, and adding to it. This makes sure that analytical workloads can handle the greatest traffic without getting too slow or missing out on growth opportunities. This article speaks about a good way to set up a cloud-first data analytics system for apps that handle money. Instead of one big data lakehouse, you should employ smaller SQL pipelines. This case study of a medium-sized fintech company shows that this architecture made it feasible to always keep an eye on compliance indicators, portfolio performance, and risk exposure, with query times of less than a second. What will happen next? Building infrastructure is cheaper, you can report crimes to the authorities more quickly, and you can modify your company model quickly. This paper talks about essential design ideas, operational benefits, and lessons gained from using SQL-driven pipelines and cloud data platforms in the real world for firms who want to improve their financial analytics infrastructure.

Keywords - Real-Time Analytics, Financial Data Integration, SQL Pipelines, Data Hubs, Cloud Architecture, Data Lakehouse, Streaming Analytics, DataOps, ETL/ELT, FinTech Infrastructure.

1. Introduction

In today's fast-paced world of finance, it's important to be able to get information and act on it very away. Banks and other financial institutions have to make swift decisions based on information. This is true whether they are watching for sudden changes in the market, reviewing their credit risk all the time, or making sure they follow the rules as they change. From trading desks to risk management teams, real-time analytics is becoming the most critical tool for staying ahead of the competition. If your insights are wrong or linger a long time, you could miss out on opportunities, lose money, or get in problems with the law. The need for quick, reliable, and useful information has grown as the financial services industry has gone digital. The big change is largely because there is now so much data available in financial markets and services. Every second, stock exchanges, payment gateways, mobile banking apps, customer interactions, IoT-enabled devices, and third-party APIs all send reinforcing data. There is a lot of new information being added to existing transactional data, such as social sentiment, behavioral analytics, and geographic indicators. This helps us understand how individuals act and how the market operates better. Getting the data is hard, but so is using it, analyzing it, and judging it in real time.

Because they are constructed on rigid and separate infrastructures, legacy systems aren't ready for this degree of complexity yet. These systems are mostly used for batch processing, but they have issues with latency, can't be expanded horizontally, and users may have to do a lot of work to add new data sources or reporting needs. Because they are all one piece, they don't operate well with the tools and services that are accessible right now. This makes it harder for personnel from different departments to work together and keep up with what's going on. since of this, businesses who only use old infrastructure often have to wait since they can't keep up with how quickly the financial sector evolves. Cloud-native architectures are a new way to think about data infrastructure that takes into account how well things work together, how easy it is for them to develop, and how adaptable they are. The basic notion behind this new style of thinking is cloud-based data hubs. They are flexible storage spaces that keep data from all throughout the enterprise. These hubs make it easier to collect relevant data by isolating storage from compute, allowing streaming, batch, and event-driven workloads to run at the same time, and enabling a shared-data paradigm. Cloud data hubs can be as big or as small as you need them to be, and they work well with many different analytics, AI/ML, and business intelligence technologies.

SQL pipelines are built into these data centers. You can edit these processes to use declarative SQL logic to change, transfer, and improve data. With SQL pipelines, businesses can keep using SQL and still enjoy the benefits of cloud-native execution engines like BigQuery, Snowflake, or Databricks. These pipelines take care of the entire ETL/ELT process on their own. They take data from these operating systems, change it in memory & then make it available for examination in a matter of seconds. They can work in both streaming & batch modes, which means they can do tasks that need to be done very instantly, including identifying fraud, keeping an eye on liquidity, or getting in touch with their certain clients.

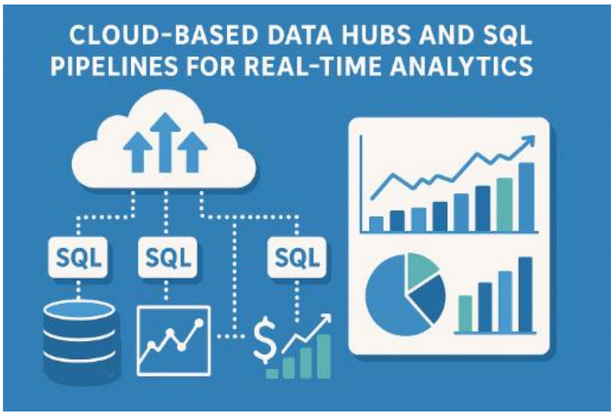

Figure 1. Cloud-Based Data Hubs and SQL Pipelines for Real-Time Analytics

The objective of this essay is to figure out the best approach to employ actual time financial analytics by putting SQL-driven pipelines & cloud data hubs together into one solution. We want to help banks & many other financial institutions, from small fintech startups to huge enterprises, make their analytics systems better so they can work more safely, respond more swiftly, and grow more wisely. We help you by giving you important information, technical help, and examples from the real world. The essay talks about the most critical pieces of a modern cloud-based analytics architecture, the benefits of a modular pipeline design, and the best strategies to make Socrates, governance, and dependability high. This article is about a strategy to create a financial analytics system that lets business users make decisions in real time, handle more data, and get insights when they need them. It shows banks and other financial institutions how to stop utilizing outdated systems and start using cloud technology's power, flexibility, and smartness.

2. Cloud-Based Data Hubs in Financial Systems

Financial firms need cloud-based data centers that are flexible, scalable, and speedy since decisions may be made rapidly and data environments are complicated. These hubs do more than simply store data; they also execute the things that existing financial analytics pipelines need to do, such as collecting, storing, updating, accessing, and sharing data. This part speaks about how to build cloud data hubs, how they are different from conventional data warehouses, which platforms are most typically used to make this change happen, and what the most common architectural features are.

2.1. What it means and what its main parts are

A cloud-based data hub is a central system in a cloud-native environment that makes it simpler to gather, organize, and share structured and semi-structured data from many various places, both within and outside the company. Data hubs can do more than merely store data for processing. They make it easier to exchange data, stream it, get to it from other sites, and combine it in other ways. This gives companies a flexible approach to look at data.

Some important parts of a data hub are:

- Layer of Metadata: This layer shows you where the data came from, how it is arranged, how it got there, how good it is, and who can view it. This makes it easy to understand. Metadata assists with compliance by making it simpler to keep track of, discover, and automatically enforce rules in financial systems.
- Storage Abstraction: The storage layer keeps data separate from apps and tools for analyzing it. This makes it easy to use flexible storage options like columnar formats (like Parquet and ORC) and object stores (like Amazon S3 and Google Cloud Storage). This abstraction allows systems to evolve on their own and in a way that saves money. They can still deal with both raw and processed data.
- Integration Layer: This element takes data (in real time or in batches), alters it (ETL/ELT), and talks to operational systems, APIs, third-party data providers, and cloud services that the company doesn't own. It links systems that handle transactions with systems that analyse data.

Together, these layers create a shared foundation for consistent, secure, and scalable data consumption across business unitsfrom risk analytics and regulatory reporting to customer segmentation and portfolio management.

2.2. Advantages over Traditional Data Warehouses

For a long time, banks and other financial institutions have found data warehouses to be quite helpful for learning about their businesses. But in today's world, where there is a lot of data, they usually have a lot of problems. Cloud-based data centers solve these problems by giving you a solution that lasts longer and can be changed to fit your needs.

- Scalability: It's typical to set up data warehouses by hand, and their hardware capacity is fixed, which makes it hard and expensive to make them bigger. Cloud data hubs, on the other hand, are flexible by design. This implies that companies may automatically add or take away computational and storage resources dependent on how much work they have to complete. This maintains performance the same, no matter how many queries or how much data there is. This is a key aspect of systems that check for fraud or perform analytics on high-frequency trading.
- Flexibility: In a world where rules and money are always changing, it's really important to be able to adapt. Adding new data sources or changing the schemas in older warehouses takes me a long time. Cloud data hubs are adaptable because they can read schemas, handle several sorts of data and pipelines, and divide up computing. This makes it simpler for data scientists and analysts to work with data that is essentially real-time and make changes swiftly without having to deal with IT procedures or predefined data formats.
- Cost-Efficiency: Cloud data centers are cheaper since they only consume resources when you need them and charge you for what you need. You can keep cold data in cheap storage systems and hot data in high-performance tiers. Cloud providers also offer tiered storage, spot instances, and full data lifecycle management, which can help you save space. This is especially important for banks and other financial institutions that handle millions of transactions and keep track of them.

2.3. Common Architectures

Several architectural models have evolved to structure cloud-based data hubs for financial use cases. Two primary approaches stand out:

2.3.1. Lakehouse Model

Data lakes are great for storing a lot of data, but they aren't very organized or reliable. Data warehouses are good at putting data in order, but they don't scale very well. The data lakehouse is the finest of both worlds. It provides teams a location to save both raw data and tables that are ready to be looked at. Lakehouses are places where you can do both business intelligence tasks and strong machine learning. This cuts down on redundant data, speeds up pipelines, and makes it simpler to construct models that can change rapidly. Financial companies employ lake houses to keep records of compliance, raw historical tick data, and credit rating algorithms. They also provide curated datasets for dashboards that show real-time data and for government audits. Two platforms employ this method: Databricks and Delta Lake. They enable you run time-travel queries, ACID transactions, and streaming ingestion, all of which are very valuable in financial markets that are hard to predict.

2.3.2. Hub-and-Spoke vs. Mesh Topology

Hub-and-Spoke Model: The data hub is the main system that links the other departmental systems, which are known as "spokes." This style is easy to use and works well for programs that everyone in the firm needs to use, such forecasts or reports to the government. But that could be a problem if a lot of teams need to get to the data at the same time and by themselves. Data Mesh: A system that isn't centralized and lets domain teams own their own data. The data products of each team are their own, but other teams can find and use them through a shared catalog. This technique is better for current FinTech ecosystems where product, risk, and analytics teams work together but yet have some freedom. But it's tougher to put into action. It makes it possible to have a federated governance system and pushes people to be innovative and flexible in some areas.

2.4. Leading Platforms

Several cloud platforms have become leaders in the financial sector due to their performance, flexibility, and ecosystem support**.**

- Snowflake: Many people believe that Snowflake is easy to use and has a decent structure for several clusters. Also, it makes it easier to automatically scale, move data safely, and work with data that isn't fully structured. It contains capabilities like zero-copy cloning and virtual warehousing that make it a great tool for banks, insurance businesses, and wealth management companies to use together for data analysis.
- Databricks: Databricks is based on Apache Spark and Delta Lake, and it can conduct many different types of data engineering and machine learning work. It's great for complicated analytics like fraud detection, portfolio optimization, and credit modeling because it has a lakehouse design and collaborative notebooks.
- Google BigQuery: A data warehouse that doesn't need a server and can search through petabyte-sized databases and undertake complex searches. FinTech companies and digital banks who care about speed and flexibility will find it interesting because it works with Looker and Vertex AI and lets you do federated searches.
- AWS Redshift: Amazon Redshift is quick and can grow with your needs. It also has a lot of connections in the AWS ecosystem. It has Redshift Spectrum to get to S3 data and AQUA to make things go faster. This is fantastic for businesses that already utilize AWS and want to keep their data organized.

3. SQL Pipelines: Engine of Real-Time Data Transformation

It has grown tougher to understand and control how data transfers across platforms as more consumers expect to be able to obtain financial information right now. The SQL pipeline is a very important part of how data works nowadays. It lets you filter, change, and interpret raw data so that you may study it. SQL pipelines help firms keep their data flowing smoothly, automate the logic behind processing data, and speed up the analytics lifecycle in cloud-based financial analytics. This section speaks about SQL pipelines, how they function, the technology that makes them work, and why they are so vital for financial systems.

3.1.1. What are SQL pipelines?

SQL pipelines are sets of SQL commands that run on their own and receive, change, and output data so that it may be used later in machine learning or analytics. Most of the time, these pipelines are built using declarative SQL reasoning. This enables data engineers and analysts to say what changes they want to see, but not how they want them to happen. SQL pipelines give you an abstraction that makes sure things can be done again, that they are done right, and that they are done quickly. This makes it easy to work with tough data.

3.1.2. Declarative Transformation Logic

In traditional programming models, data transformations are written imperatively, requiring users to specify control flow, data iteration, and execution order. SQL pipelines, by contrast, use declarative logicstatements like SELECT, JOIN, WHERE, and GROUP BY to describe data manipulations in a concise and human-readable format. This makes it easier for technical and non-technical stakeholders to work together, see what's going on, and audit things. This is particularly crucial in regulated fields like banking.

3.1.3. Streaming SQL vs. Batch SQL

One important difference between SQL pipelines is how they are executed:

Batch SQL works with data at specified intervals, usually from snapshots or static datasets. This mode is best for regulatory reporting, risk simulations, or reconciling at the end of the day. SQL Streaming You can view data almost straight away since it works with data streams that are continually flowing in. This is very important for swiftly verifying credit, keeping an eye on discounts, and discovering fraud. You can use the same SQL logic on data streams that are occurring right now with Apache Flink and Google Cloud Dataflow. They achieve this automatically by employing windowing mechanisms and processing events as they happen. Both modalities may function together in a pipeline structure, offering financial companies the freedom to choose the right balance between performance and cost.

3.2. Parts of SQL Pipelines

A common SQL pipeline has many steps that rely on each other and work together to manage the lifecycle of data from source to insight:

3.2.1. Taking in

The intake layer collects data from a lot of different locations, including as APIs, message brokers (like Kafka or Pub/Sub), cloud storage, and services that aren't yours. ACH transactions, market data streams, SWIFT communications, or telemetry from financial applications are all instances of this. The ingestion phase has to be able to rapidly figure out the schema, get rid of duplicates, and transfer the data to the pipeline.

3.2.2. Parsing

You can't ingest raw data unless you organize it into things like JSON objects or tables. Parsing is the process of verifying what kind of data it is, altering it, and making it standard. This step is very important for financial analytics since data might arrive in many various formats, such as CSV, XML, Avro, and others. and fields may be nested or encoded. Parsing also makes timestamps more consistent, which is highly crucial for getting accurate results when looking at time series.

3.2.3. Transformation

The transformation stage applies SQL logic to shape the data into usable forms. This includes things like filtering and deduplication.

- Putting things together and breaking them apart by time
- Combining datasets, such transactions with reference data
- Using external mappings (like currency conversion) to improve data
- Finding derived measures, such moving averages and volatility indices

In financial systems, transformations generally include combining transactional data with customer, compliance, or market data to provide meaning to the information.

3.2.4. Orchestration

Orchestration takes care of scheduling, keeping track of dependencies, and deciding what order the various parts of the pipeline should operate in. This includes fixing errors, trying again, filling in gaps, and sounding alerts. You may use Apache Airflow, Prefect, and Dagster to keep a watch on SQL pipelines and make sure they can handle problems. Orchestration makes sure that any datasets that need to be updated receive them right immediately, in real time.

3.3. SQL Engines and Frameworks

To help with the large-scale construction and execution of SQL pipelines, a number of new tools and frameworks have come out:

- Dbt (Data Build Tool): Dbt is an important tool for analytics engineering since it enables you to split SQL transformations into smaller chunks and keep track of different versions. It generates dependency graphs from SQL models, loads data in little pieces, and makes it easy to write documentation and run tests. In finance, dbt is a standard way to construct data models that obey the rules. This makes it easier to see where the money and risk analytics went and where they went.
- Apache Beam: You can use Apache Beam to work with both batch and streaming data in one programming model. Teams may use Beam to build SQL code that explains how real-time data works and run it on several platforms at once. It does this with the use of SQL extensions and runners, such as Google Cloud Dataflow.
- Flink SQL: The SQL engine in Apache Flink is only for viewing data as it comes in. You can use ordinary SQL to deal with events as they happen, repair things that aren't working well, and join tables on the fly. This is especially useful for financial tasks that need to be done quickly, like keeping an eye on deals, grading transactions, and portfolios.
- Spark SQL: Spark SQL has a SQL engine that can handle a lot of data and is spread out. It is typically used with structured streaming. When it comes to transformations that involve a lot of processing capacity, including stress testing, scenario analysis, and training historical models, Spark SQL is the ideal solution for financial analytics.

Each framework offers unique advantages, and the choice often depends on latency requirements, data volumes, and the surrounding ecosystem.

3.4. Role in Financial Analytics

Modern financial analytics operations depend on SQL pipelines to speed up, make data more accurate, and give it meaning across all data domains. You may witness their effect in a few important areas:

- Standardization: When it comes to money, basic data may not always be in the appropriate shape. For instance, it could have to work with multiple types of money, different methods to format timestamps, and different ways to name variables. SQL pipelines convert this data into common schemas, which makes it easier to add up and compare data from multiple locations, times, and kinds of things.
- Enrichment: Enrichment is the process of adding more information from reference databases to raw data to make it better. You can add information about the instruments, the ratings of the issuers, or the risk categories to the trade data. SQL pipelines make this process easier by making sure that any future analytics are based on the right context and are ready to be used to make decisions.
- Joining Across Systems: You usually need data from a lot of different systems to get one financial insight. For example, you could combine loan applications from a CRM with payment history from a core banking system. SQL pipelines make it easy to mix data from different systems in ways that would be hard or take a long time to do by hand. These links let you do important things like get a complete picture of a client, find out how much money they have at risk, and report on their overall balance.

4. Real-Time Financial Analytics: Architecture and Challenges

Algorithmic trading, digital banking & AI-powered decision-making are all hot right now, so financial businesses need to know how to execute analytics in real time. Banks & other financial organizations need systems that can rapidly & accurately give them legal information. This way, they can keep a watch on the market, see how much money they have, & provide each consumer with the best recommendations. These kinds of systems are still challenging to make. It needs a smart mix of cloud-native tools, data streaming frameworks, & good design rules. It also needs to fix issues with operations, compliance, & architecture at the same time. This part talks about the basic needs of real-time financial analytics, looks at different design options including event-driven architectures, compares processing models, & talks about the biggest problems that come up when trying to make these ideas work.

4.1. Requirements for Real-Time Finance

In finance, real-time analytics means more than just being able to perform queries rapidly. The basic structure has to follow a lot of severe standards that the company needs to follow in order to stay in business and stay within the law.

- Low-Latency Ingestion: The best thing about real-time analytics is that it can look at data straight away. The analytics engine needs to get things like transaction events, portfolio changes, fraud alerts, and market feeds in less than a second. Payment gateways, stock exchanges, and risk assessment systems provide banks and other financial institutions data that changes fast and all the time. We need communications layers with low latency, like Apache Kafka or AWS Kinesis, and ingestion services that can expand to keep data up to date and make things go faster.
- Consistency and Correctness: Speed doesn't matter if you can't trust it. All systems, whether they are used for regulatory reporting or by clients, must contain the same and precise financial information. Real-time pipelines need to use strong consistency models, make sure that schemas are followed, and make sure that messages are transmitted even if nodes or the network go down. You also need to use watermarking and event-time processing to deal with events that happen in the wrong sequence, such transaction logs that come in late.
- Auditability and Compliance: The financial industry needs to follow a number of strict rules, such as SOX, PCI DSS, MiFID II, and Basel III. To meet these criteria, you need to be honest, know where things are, and keep old data up to date. Real-time systems need to keep track of all changes, make sure the data lineage is clear, and make it simple to use replay techniques to go back to earlier states. This implies that the layers that protect, alter, and process data all need to work together quickly.

4.2. Data design based on events

Using event-driven architecture (EDA) is one of the finest approaches to create systems that can look at financial data in real time. In this model, any change in the business environment, whether a transaction, an ATM withdrawal, or the acceptance of a loan, is an event that initiates activities immediately away.

4.2.1. Kafka, Kinesis, and Pub/Sub Systems

To support event-driven design, messaging systems act as the central nervous system for event propagation. Apache Kafka is the de facto standard for distributed streaming. It has high throughput, fault tolerance, and message replay. Real-time trading systems, credit rating, and compliance warnings are some examples of financial use cases. AWS Kinesis is a service that lets you ingest, handle, and analyze streaming data on AWS. Kinesis works well with other AWS analytics tools like Lambda and Redshift, which makes it perfect for cloud-native finance stacks. Google Cloud Pub/Sub is a communications service that works all over the world and can be scaled up or down to separate systems and let real-time pipelines run across regions. Pub/Sub’s event delivery guarantees make it well-suited for customer engagement and fraud analytics in banking apps. These systems ensure asynchronous, decoupled, and reactive workflowswhere data flows are triggered automatically and processed in parallel, increasing both speed and fault tolerance.

4.3. Streaming vs. Micro-Batch Models

When implementing real-time data flows, teams must choose between two primary models: streaming and micro-batch. While both aim to reduce latency, they differ in execution granularity and system complexity.

4.3.1. Streaming Processing

Right now, streaming data shows that it is only processing one event at a time. This helps you make decisions in less than a second, which is great for really important things like keeping an eye on deals, stopping money laundering (AML), or figuring out how much risk there is. Apache Flink and Kafka Streams are two technologies that achieve this quite well. They have great event processing (CEP), windowing capabilities, and semantics that make sure that things only happen once.

Good points:

- Very low latency
- Fine control over the timing and sequence of events
- Great for keeping an eye on risks all the time and sending out compliance notifications
- Cons: It's hard to run.
- Needs careful tweaking to balance throughput and latency

4.3.2. Processing in Micro-Batches

This approach collects data in little groups throughout time, such every 5 to 30 seconds, and then analyzes it all at once. Frameworks like Apache Spark Structured Streaming and Snowflake's Snowpipe employ micro-batch mode. This makes it simple to utilize and stay up to date with business intelligence tools that come after them.

Pros:

- Easier to check and fix
- Easier to work with older batch systems
- Good for dashboards and ETL that are almost real-time

Disadvantages:
- A little more delay (around a second)
- Not good for ultra-high-frequency usage scenarios

In fact, a lot of companies use a hybrid strategy, with streaming for real-time alerts and micro-batches for reporting and aggregation.

4.4. Common Obstacles
Even while real-time analytics sounds great, putting these kinds of systems into place on a large scale is hard for both technological and organizational reasons.

4.4.1. Data Silos
A lot of banks and other financial institutions employ different systems for things like core banking, managing customer relationships (CRM), the treasury, and managing risk. Usually, these systems store data in different ways, are run by separate organizations, and need to be updated at various times. This makes it challenging to get a precise, current picture of how consumers behave or how harmful a business is.
- Some such solutions are:
- Setting up data hubs with schema registries

Using CDC (Change Data Capture) solutions like Debezium to keep databases that are not connected to each other in sync Using data mesh principles to set up cross-domain ownership.

4.4.2. Time delay Hiccups
Getting low latency from start to finish involves more than simply rapid streaming; it has to be optimized at every level, such as:
- Slow writes to the database that are behind the ingest rates
- Joins that don't work well across big datasets
- Too few computing clusters at busy times
- To get around these, teams use:
- Computing in memory (like Apache Ignite)
- Materialized views for results that have already been grouped

Orchestration solutions that automatically scale, such Kubernetes or serverless engines

4.4.3. Management Trade-offs
Real-time systems frequently have trouble finding the right balance between data flexibility and control. Developers demand schemas that can be changed and deployed quickly, while compliance teams require version control, lineage tracking, and access limitations.
- Role-based access control (RBAC) and data masking are two important options.
- Integration with metadata systems like Amundsen or Collibra CI/CD for versioning pipelines with tools like dbt and GitOps
- Governance is not optional in financial settings; it must be built into the very structure of real-time pipelines.

5. Case Study: Building a Real-Time Risk Monitoring Platform
5.1. Background
Risk management systems change quickly because the financial sector is always changing. This case study is on a medium-sized financial services company that does institutional trading, retail banking, and wealth management. The company found that traditional risk analytics, which only looked at data from the end of the day or the hour, were not enough to protect its assets and follow the rules as it grew its client base and product line. Leaders said that there should be a system that can keep an eye on risk in real time so that fraud can be found, risk can be assessed throughout the day, and compliance can be ensured. The compliance and risk management teams need to be able to quickly see position exposures, counterparty activity, and liquidity ratios. The fraud and internal audit teams wanted to keep an eye on strange transaction patterns that may have gone unnoticed for a long time. The company started a six-month project to upgrade its data and analytics infrastructure using cloud-native tools, streaming technologies, and modular SQL-based pipelines in order to reach these goals.

5.2. What the Problem Is
The organization's present architecture includes old relational databases and ETL pipelines that can only handle data in batches. The pipelines transferred data from important banking, trade execution, and CRM systems to a specific data storage

location on the site every hour. The lengthy latency was long enough to make a report, but it hurt high-frequency trading, fraud detection, and compliance monitoring.

The main limits were:

- Scalability Limitations: The data warehouse had trouble handling more than a certain number of users and queries at the same time, especially when the market was closed or reports were due.
- High Latency: Hourly updates weren't enough to find and deal with quick changes in risk exposure, counterparty defaults, or suspicious trading activity.
- Isolated Systems: Trading desks, credit engines, and customer systems all used different platforms, thus it was impossible to get a full, real-time profile.
- Manual Intervention: Risk analysts often relied on spreadsheets or customized scripts to provide specific risk insights, increasing friction and exacerbating human error.

The commercial risk was clear: putting off insights may lead to penalties for not following the rules, delays in stopping fraud, and, in the end, a loss of trust from customers.

5.3. System Design

To get around these limits, the architecture team built a cloud-native, event-driven platform that can take in, change, and show data in real time.

5.3.1. Basic Design Principles:

- First in the Cloud: Use a fully managed, cloud-native design to get rid of infrastructure problems and make your system more scalable.
- Architecture of the Lakehouse: Add both raw and refined data to a lakehouse structure to make it more flexible and cut down on duplication.
- SQL-Driven Pipelines: Use declarative SQL pipelines for transformations to make it easier to track changes, see what's going on, and speed up development.
- Streaming-Enabled: Add a communications architecture with a lot of bandwidth so that events (such trades and payments) may be processed in real time.

5.3.2. Technology Stack:

We picked Snowflake as the main data warehouse and lakehouse layer because it separates compute and storage, works with semi-structured data, and has built-in scalability features.

- The SQL-based ELT layer was made possible by dbt (Data Build Tool), which let teams transform and model data over time while keeping track of changes, testing, and writing down what they did.
- Apache Kafka took care of the real-time streaming of payment logs, trade events, and application telemetry.
- Change Data Capture (CDC) from transactional systems was done using Fivetran and Airbyte, which synced changes in a matter of seconds.

Power BI and Looker were set up to show dashboards and provide alerts in real time, with data sources connected directly to Snowflake.

5.4. Important Implementation Highlights

The installation took place in phases, starting with intake and ending with transformation and visualization.

5.4.1. CDC and Consumption

The trade execution engine and the main banking system both have CDC interfaces. These connections watched for changes, additions, and deletions in almost real-time and delivered records of those changes to Kafka topics. Kafka consumers utilized Snowpipe to move event streams to Snowflake staging tables, where they were kept for later use. The CDC structure lets the organization keep up-to-date records of transactions, changes in position, and contacts with customers without putting too much strain on source systems.

5.4 2. SQL-based ELT with dbt

Data engineers produced more than 75 modular dbt models for trade normalization. These models were grouped by product, geography, and currency.

- Finding risk measures (such Value at Risk and exposure limitations)
- Counterparty profiling, which means putting together transactions with CRM and credit assessment data
- Assessing the likelihood of fraud by using heuristics and transaction speed

We tested, documented, and version-controlled every dbt model. This made it possible to repeat the process and check it, which is an important need for internal compliance audits.

5.4.3. Dashboards in Real Time

Both Power BI for operations teams and Looker for compliance and analytics teams made dashboards. The main features were:

- Heat maps that show exposure by asset class
- Widgets that let you know about fraud in real time
- Alerts on breaches of risk limits throughout the day
- In-depth breakdowns by account, location, and kind of transaction

Notifications were set up to let risk officers know via Slack and email when specific limits were broken (for example, when the exposure limit was more than 95%).

5.5. The outcome

The switch to real-time architecture brought about huge operational and strategic advantages for all departments.

5.5.1. Improvements in Performance:

- Reducing Latency: The average freshness of the most relevant risk datasets went from 45 minutes to less than 3 minutes.
- Scalability: The firm can presently handle more than 10,000 streaming trade events per second without any problems during peak trading times.
- Query Speed: Snowflake's multi-cluster compute architecture made it possible to update dashboards and ask about risks at the same time without slowing things down.

5.5.2. Effects on the economy:

- Better at detecting fraud: Faster data entry and better risk models made it easier to find signs of fraud quickly, especially for transactions that happen a lot and strange account behavior.
- Regulatory Preparedness: Compliance reporting became more efficient and clear thanks to real-time audit logs, lineage tracking in dbt, and replay features.
- Analyst Output: Risk analysts cut down on data preparation time by more than 60% by using automated transformations and easy-to-use dashboards. This lets them focus on strategic modeling and scenario analysis.

6. Conclusion

Banks and other financial institutions are redefining how they do real-time analytics by combining SQL-driven pipelines with cloud-based data storage. Businesses can now take in, process, and analyze data faster and more efficiently than ever before, thanks to scalable lakehouse designs, declarative SQL operations, and decoupling storage from processing. These new technologies fix a lot of the drawbacks that older systems have. They help you access information quickly, make things easier to do, and always keep an eye on risk and compliance. From a strategic point of view, there are a lot of good things that can happen. Cloud-native data hubs help banks communicate and aggregate data from different departments in a single and flexible way. SQL pipelines take raw data and make it more useful and easy to utilize. You can use this information to do things like check your credit score, look for fraud, manage your portfolio, and get real-time updates on consumers. They all assist people make quick decisions, which makes it easier to deal with new dangers and chances. It's not only a technical feat to get real-time insight; it's something that businesses need to do right now because the market is becoming less stable, there are more rules and regulations, and customers demand more.

Companies who still use old, slow technology are falling behind more and more. They have to wait longer to get things done because they can't access reliable, up-to-date information. This makes it tougher to be open-minded, trust other people, and come up with new ideas. According to this report, financial analytics need to be better by employing SQL-based processing and cloud-based technologies. There are adjustments that can be made that are both possible and helpful, as shown by a look at best practices in architecture and a real-life case study. If financial businesses want to get better at analytics, they need to have a clear approach. They should buy technologies that help them see real-time information for all of their alternatives, develop modular and controlled data pipelines, and employ cloud technology. Getting rid of obsolete batch systems and developing a new financial analytics system that works in real time and is ready for the future is highly crucial.

References

[1] Poojara, Shivananda, et al. "Serverless data pipelines for IoT data analytics: A cloud vendors perspective and solutions." *Predictive Analytics in Cloud, Fog, and Edge Computing: Perspectives and Practices of Blockchain, IoT, and 5G*. Cham: Springer International Publishing, 2022. 107-132.

[2] Dutta, Kamalika, and Manasi Jayapal. "Big data analytics for real time systems." *Big Data analytics seminar*. 2015.

[3] Mishra, Sarbaree, and Sairamesh Konidala. “Automated Data Mapping and Schema Matching For Improving Data Quality in Master Data Management”. *International Journal of Emerging Trends in Computer Science and Information Technology*, vol. 4, no. 3, Oct. 2023, pp. 80-90

[4] Guntupalli, Bhavitha. “ETL Architecture Patterns: Hub-and-Spoke, Lambda, and More”. *International Journal of AI, BigData, Computational and Management Studies*, vol. 4, no. 3, Oct. 2023, pp. 61-71

[5] Yaganti, Dheerendra. "Leveraging .NET for Real-Time Big Data Analytics and Decision Support Systems." *European Journal of Advances in Engineering and Technology* 8.2 (2021): 155-160.

[6] Shaik, Babulal, Jayaram Immaneni, and K. Allam. "Unified Monitoring for Hybrid EKS and On-Premises Kubernetes Clusters." *Journal of Artificial Intelligence Research and Applications* 4.1 (2024): 649-669.

[7] Manda, J. K. "Data privacy and GDPR compliance in telecom: ensuring compliance with data privacy regulations like GDPR in telecom data handling and customer information management." *MZ Comput J* 3.1 (2022).

[8] Koppad, Saraswati, et al. "Cloud computing enabled big multi-omics data analytics." *Bioinformatics and biology insights* 15 (2021): 11779322211035921.

[9] Lalith Sriram Datla, and Samardh Sai Malay. “Patient-Centric Data Protection in the Cloud: Real-World Strategies for Privacy Enforcement and Secure Access”. *European Journal of Quantum Computing and Intelligent Agents*, vol. 8, Aug. 2024, pp. 19-43

[10] Immaneni, J., & Salamkar, M. (2020). Cloud migration for fintech: how kubernetes enables multi-cloud success. *International Journal of Emerging Trends in Computer Science and Information Technology*, *1*(3), 17-28.

[11] Mishra, Sarbaree, et al. “Hyperfocused Customer Insights Based On Graph Analytics and Knowledge Graphs”. *International Journal of AI, BigData, Computational and Management Studies*, vol. 4, no. 4, Dec. 2023, pp. 88-99

[12] Chen, Weisi, et al. "Real-time analytics: Concepts, architectures, and ML/AI considerations." *IEEE Access* 11 (2023): 71634-71657.

[13] Shaik, Babulal. "Developing Predictive Autoscaling Algorithms for Variable Traffic Patterns." *Journal of Bioinformatics and Artificial Intelligence* 1.2 (2021): 71-90.

[14] Nookala, G., Gade, K. R., Dulam, N., & Thumburu, S. K. R. (2024). Post-quantum cryptography: Preparing for a new era of data encryption. *MZ Computing Journal*, *5*(2), 012077.

[15] RABHI, FETHI A., and ANDREW BERRY. "Real-Time Analytics: Concepts, Architectures, and ML/AI Considerations."

[16] Allam, Hitesh. "Bridging the Gap: Integrating DevOps Culture into Traditional IT Structures." *International Journal of Emerging Trends in Computer Science and Information Technology* 3.1 (2022): 75-85.

[17] Guntupalli, Bhavitha, and Surya Vamshi ch. “Designing Microservices That Handle High-Volume Data Loads”. *International Journal of AI, BigData, Computational and Management Studies*, vol. 4, no. 4, Dec. 2023, pp. 76-87

[18] Lalith Sriram Datla, and Samardh Sai Malay. “Data-Driven Cloud Cost Optimization: Building Dashboards That Actually Influence Engineering Behavior”. *Los Angeles Journal of Intelligent Systems and Pattern Recognition*, vol. 4, Feb. 2024, pp. 254-76

[19] Patel, Piyushkumar. "The Implementation of Pillar Two: Global Minimum Tax and Its Impact on Multinational Financial Reporting." *Australian Journal of Machine Learning Research & Applications* 1.2 (2021): 227-46.

[20] Selvarajan, Guru Prasad. "Optimising Machine Learning Workflows in SnowflakeDB: A Comprehensive Framework Scalable Cloud-Based Data Analytics." *Technix International Journal for Engineering Research* 8.11 (2021).

[21] Mishra, Sarbaree. “Incorporating Automated Machine Learning and Neural Architecture Searches to Build a Better Enterprise Search Engine”. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, vol. 4, no. 4, Dec. 2023, pp. 65-75

[22] Balkishan Arugula, and Vasu Nalmala. “Migrating Legacy Ecommerce Systems to the Cloud: A Step-by-Step Guide”. *Los Angeles Journal of Intelligent Systems and Pattern Recognition*, vol. 3, Dec. 2023, pp. 342-67

[23] Rosandic, Josip. *Real-time streaming data management, processing, analysis and visualisation*. Diss. PhD thesis, University of Zagreb, 2022.

[24] Mohammad, Abdul Jabbar. “Dynamic Labor Forecasting via Real-Time Timekeeping Stream”. *International Journal of AI, BigData, Computational and Management Studies*, vol. 4, no. 4, Dec. 2023, pp. 56-65

[25] Mohammad, Abdul Jabbar. “Chrono-Behavioral Fingerprinting for Workforce Optimization”. *International Journal of AI, BigData, Computational and Management Studies*, vol. 5, no. 3, Oct. 2024, pp. 91-101

[26] Mohna, Hosne Ara, et al. "AI-ready data engineering pipelines: a review of medallion architecture and cloud-based integration models." *American Journal of Scholarly Research and Innovation* 1.01 (2022): 319-350.

[27] Mishra, Sarbaree. “The Lifelong Learner - Designing AI Models That Continuously Learn and Adapt To New Datasets”. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, vol. 5, no. 1, Mar. 2024, pp. 68-78

[28] Jani, Parth. "Predicting Eligibility Gaps in CHIP Using BigQuery ML and Snowflake External Functions." *International Journal of Emerging Trends in Computer Science and Information Technology* 3.2 (2022): 42-52.

[29] Nookala, G. (2023). Microservices and Data Architecture: Aligning Scalability with Data Flow. *International Journal of Digital Innovation*, *4*(1).

[30] Manda, Jeevan Kumar. "Zero Trust Architecture in Telecom: Implementing Zero Trust Architecture Principles to Enhance Network Security and Mitigate Insider Threats in Telecom Operations." *Journal of Innovative Technologies* 5.1 (2022).

[31] Thallam, Naga Surya Teja. "Comparative Analysis of Public Cloud Providers for Big Data Analytics: AWS, Azure, and Google Cloud." *International Journal of AI, BigData, Computational and Management Studies* 4.3 (2023): 18-29.

[32] Arugula, Balkishan. "AI-Powered Code Generation: Accelerating Digital Transformation in Large Enterprises". *International Journal of AI, BigData, Computational and Management Studies*, vol. 5, no. 2, June 2024, pp. 48-57

[33] Jani, Parth, and Sarbaree Mishra. "Governing Data Mesh in HIPAA-Compliant Multi-Tenant Architectures." *International Journal of Emerging Research in Engineering and Technology* 3.1 (2022): 42-50.

[34] Veluru, Sai Prasad. "Threat Modeling in Large-Scale Distributed Systems." *International Journal of Emerging Research in Engineering and Technology* 1.4 (2020): 28-37.

[35] Allam, Hitesh. "Declarative Operations: GitOps in Large-Scale Production Systems." *International Journal of Emerging Trends in Computer Science and Information Technology* 4.2 (2023): 68-77.

[36] Mohammad, Abdul Jabbar, and Seshagiri Nageneini. "Temporal Waste Heat Index (TWHI) for Process Efficiency". *International Journal of Emerging Research in Engineering and Technology*, vol. 3, no. 1, Mar. 2022, pp. 51-63

[37] Chaganti, Krishna C. "Leveraging Generative AI for Proactive Threat Intelligence: Opportunities and Risks." *Authorea Preprints*.

[38] Shaik, Babulal. "Automating Compliance in Amazon EKS Clusters With Custom Policies." *Journal of Artificial Intelligence Research and Applications* 1.1 (2021): 587-10.

[39] Dubuc, Timothée, Frederic Stahl, and Etienne B. Roesch. "Mapping the big data landscape: technologies, platforms and paradigms for real-time analytics of data streams." *IEEE Access* 9 (2020): 15351-15374.

[40] Lalith Sriram Datla, and Samardh Sai Malay. "From Drift to Discipline: Controlling AWS Sprawl Through Automated Resource Lifecycle Management". *American Journal of Cognitive Computing and AI Systems*, vol. 8, June 2024, pp. 20-43

[41] Talakola, Swetha. "Analytics and Reporting With Google Cloud Platform and Microsoft Power BI". *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, vol. 3, no. 2, June 2022, pp. 43-52

[42] Nookala, G., Gade, K. R., Dulam, N., & Thumburu, S. K. R. (2024). Building Cross-Organizational Data Governance Models for Collaborative Analytics. *MZ Computing Journal*, *5*(1).

[43] Abdul Jabbar Mohammad. "Timekeeping Accuracy in Remote and Hybrid Work Environments". *American Journal of Cognitive Computing and AI Systems*, vol. 6, July 2022, pp. 1-25

[44] Jaiswal, Jitendra Kumar. "Cloud Computing for Big Data Analytics Projects." (2018).

[45] Guntupalli, Bhavitha. "Data Lake Vs. Data Warehouse: Choosing the Right Architecture". *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, vol. 4, no. 4, Dec. 2023, pp. 54-64

[46] Chaganti, Krishna. "Adversarial Attacks on AI-driven Cybersecurity Systems: A Taxonomy and Defense Strategies." *Authorea Preprints*.

[47] Sreekandan Nair , S. (2023). Digital Warfare: Cybersecurity Implications of the Russia-Ukraine Conflict. International Journal of Emerging Trends in Computer Science and Information Technology, 4(4), 31-40. https://doi.org/10.63282/7a3rq622

[48] Manda, J. K. "Cybersecurity Automation in Telecom: Implementing Automation Tools and Technologies to Enhance Cybersecurity Incident Response and Threat Detection in Telecom Operations." *Advances in Computer Sciences* 4.1 (2021).

[49] Mishra, Sarbaree, and Jeevan Manda. "Improving Real-Time Analytics through the Internet of Things and Data Processing at the Network Edge ". *International Journal of Emerging Research in Engineering and Technology*, vol. 5, no. 2, June 2024, pp. 41-51

[50] Abdul Jabbar Mohammad. "Biometric Timekeeping Systems and Their Impact on Workforce Trust and Privacy". *Journal of Artificial Intelligence & Machine Learning Studies*, vol. 8, Oct. 2024, pp. 97-123

[51] Ansari, Aftab. "Evaluation of cloud based approaches to data quality management." (2016).

[52] Arugula , Balkishan. "Ethical AI in Financial Services: Balancing Innovation and Compliance". *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, vol. 5, no. 3, Oct. 2024, pp. 46-54

[53] Vasanta Kumar Tarra, and Arun Kumar Mittapelly. "AI-Driven Fraud Detection in Salesforce CRM: How ML Algorithms Can Detect Fraudulent Activities in Customer Transactions and Interactions". *American Journal of Data Science and Artificial Intelligence Innovations*, vol. 2, Oct. 2022, pp. 264-85

[54] Allam, Hitesh. "Unifying Operations: SRE and DevOps Collaboration for Global Cloud Deployments". *International Journal of Emerging Research in Engineering and Technology*, vol. 4, no. 1, Mar. 2023, pp. 89-98

[55] Saleem, Saima, and Monica Mehrotra. "Data analytics and mining: platforms for real-time applications." *Data driven decision making using analytics*. CRC Press, 2021. 61-80.

[56] Shantharajah, S. P., and E. Maruthavani. "A survey on challenges in transforming No-SQL data to SQL data and storing in cloud storage based on user requirement." *International Journal of Performability Engineering* 17.8 (2021): 703.