The Why and How of Streaming-First Data Architectures

By Kevin Petrie

April, 2020

Research sponsored by:

About the Author

Kevin Petrie has deciphered what technology means to practitioners for 25 years, as an industry analyst, writer, instructor, marketer, and services leader. Kevin launched, built, and led a profitable data services team for EMC Pivotal in the Americas and Europe, and ran field training at the data integration software provider Attunity (now part of Qlik). A frequent public speaker and author of two books on data streaming, Kevin also is a data management instructor at eLearningCurve.

About Eckerson Group

Eckerson Group helps organizations get more value from their data through research, consulting, and education. Our experts each have more than 25+ years of experience in the field, specializing in business intelligence, data architecture, data governance, analytics, and data management. We provide organizations with expert guidance during every step of their data and analytics journey. Get more value from your data. Put an expert on your side. Learn what Eckerson Group can do for you!

About This Report

This research was sponsored by Equalum. Eckerson Group does not endorse any specific vendor offerings.

Table of Contents

Executive Summary

Organizations build streaming-first architectures to increase revenue, reduce cost, and control risk. By replacing traditional batch-only ETL processes, data teams can enable real-time decision making & machine learning, improve efficiency, increase scale, and accelerate applications. Many organizations embrace streaming-first architectures as part of broader strategic initiatives that include cloud modernization and data pipeline automation.

To realize these benefits, data teams must carefully scope their use cases and select the right technologies for efficiently manipulating data in motion. They must streamline processes while still integrating heterogeneous end points and flexibly adapting their architecture to address changing business requirements.

This report examines the benefits, challenges, adoption patterns and use cases for a streaming-first approach to data management. It guides data leaders through the capabilities and planning criteria for each architectural component – data sources, collection, transformation, targets and analytics – as well as the role of on-premises, hybrid and cloud infrastructure. Readers will learn guiding principles to navigate their journey to a streaming-first data architecture.

Key Recommendations

- **Consider a streaming-first data architecture to modernize your business.** Event streaming breaks the bottlenecks of legacy batch-only ETL approaches, laying a scalable, efficient and real-time foundation for advanced analytics. By updating data incrementally at near-zero latency, a streaming-first architecture enables your business to transform and analyze more data, faster, per unit of CPU or bandwidth. You can cost-effectively address new use cases.

- **Build a holistic plan.** To realize the efficiency and analytics benefits of a streaming-first architecture, carefully define your use cases and requirements. As you design your architecture to meet those requirements, weigh the tradeoffs and interdependencies of each component.

- **Take an incremental approach.** Seek first to alleviate pain – in the form of ballooning costs or business delays – by eliminating one or more batch-processing bottlenecks with streaming solutions based on technologies such as CDC or Kafka. Once you achieve this quick win, extend your streaming architecture to new end points and start to address new use cases. As you scale, consider cloud Infrastructure as a Service (IaaS) platforms that smooth out operating costs while minimizing lock-in risk.

The Rise of Streaming-First Architectures

We live in an event-driven world. Amazon one click purchases, machinery gear turns and myriad other events stream off all types of devices, creating business opportunities in digital form.

Streaming-first data architectures enable organizations to capture that short-lived business value by processing event data as it is created and before it is stored. Organizations build streaming-first architectures to enable real-time decision making & machine learning, improve efficiency, increase scale, and accelerate applications. They increase revenue, reduce cost, and control risk.

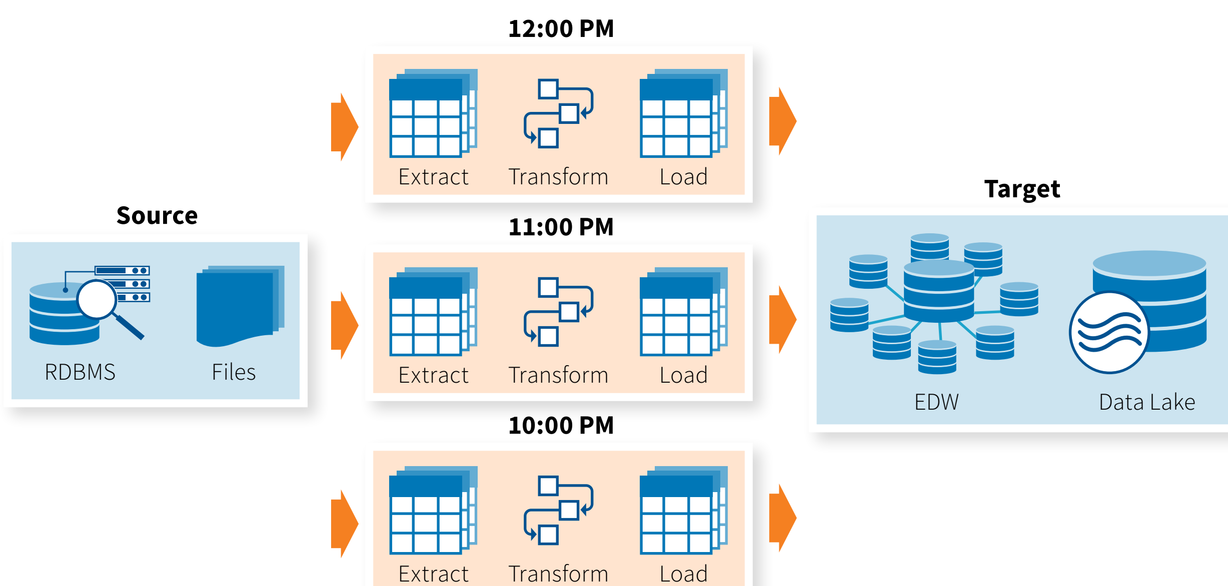
This huge opportunity spurs action across the organization. IT teams build event streams, often combining Apache Kafka and data replication solutions such as change data capture (CDC), to reduce full-load batch processing and streamline data transfer. Business and analytics teams, meanwhile, seek to analyze those event streams. They want to capitalize on events more rapidly and devise new strategies.

But challenges abound. Data teams must carefully scope their use cases and select the right technologies for efficiently manipulating data in motion. They must streamline processes while still integrating heterogeneous end points.

The Problem with Business as Usual: Batch-Only Architectures

For years enterprises processed data in periodic batches. Data teams scheduled "Extract, Transform and Load (ETL)" software to copy full batches of operational data from a finance, sales or HR database, every hour, day or week. These batches also might be collections of files.

Figure 1. Traditional Batch-Only Data Architecture



The ETL software, often running on an intermediate server, then loaded the full batch copy into a separate repository, transforming its rows and columns along the way. Organizations still use batch ETL to support lower-volume and predictable workloads for basic tasks like record-keeping and weekly reporting. Figure 1 illustrates these traditional architectures.

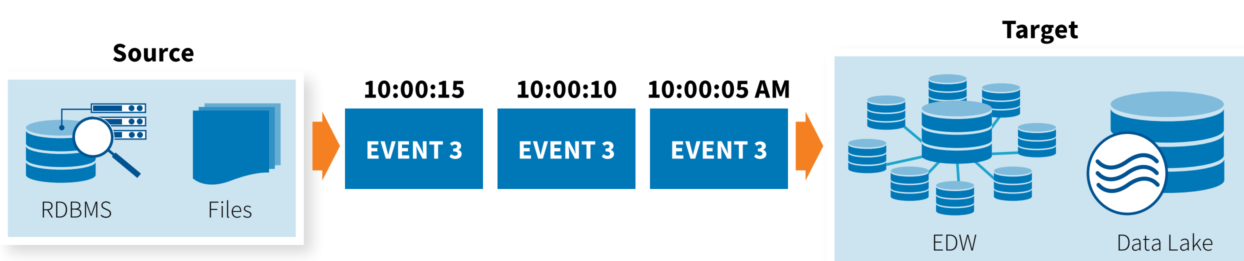
Outside this niche, “business as usual” batch-only architectures fail to meet modern enterprise requirements. They fail with latency, because many of today’s data consumers cannot wait for batch updates. Sales executives need hourly dashboard views of revenue by territory, rep or product. Factory managers need alerts of operational wear and tear in minutes. Remote healthcare providers need vital stats and charts tracking home-based patients in seconds. Credit-card companies need to identify and block suspicious transactions in milliseconds.

“Business as usual” batch-only architectures also fail to efficiently handle large datasets. They repeatedly copy unchanged data, tying up CPU cycles, gumming up memory and forcing costly hardware upgrades. Batch-only architectures consume so many resources that they cannot cost-effectively process the large data volumes required for advanced analytics use cases such as machine learning or other types of Artificial Intelligence. The batch-only approach bleeds money and blocks innovation for many modern use cases.

The Streaming-First Approach

Streaming-first architectures solve the problems of speed and efficiency. First is speed. Wherever possible, they “stream” live events from source to target, often transforming these events along the way. This synchronizes datasets and enables insights on a real-time or near real time basis. This also solves the problem of efficiency. By processing incremental changes continuously, they eliminate the need for repeated batch replication of unchanged data. This drastically reduces processing power and bandwidth requirements, helping organizations scale to support higher data volumes with the same infrastructure. Figure 2 illustrates a sample data streaming architecture at a conceptual level.

Figure 2. Streaming Architecture



Business Benefits of a Streaming-First Architecture

Data teams can use event streaming to more easily execute projects and meet service-level requirements. They can add data sources, absorb rising data volumes, and improve application response times without adding infrastructure.

Real-time decision making. Streaming-first architectures feed data real-time into platforms such as data warehouses and data lakes. This gives business analysts the instant insights needed to respond to short-lived opportunities and risks. A retail analyst can identify customer purchase patterns the morning of Cyber Monday, then adjust prices that afternoon. Data scientists and analysts also use streaming analytics to build new layers of understanding for a business problem. For example, the retail analyst uses results of her Cyber Monday campaign to enrich her forecast and strategy for the coming year.

New analytical use cases. Event streaming enables enterprises to create opportunities and solve or prevent problems. Proactive maintenance and fraud detection use real-time event streaming to control cost and risk. A fleet operator can dispatch a technician to check warning lights before a delivery truck breaks down. A credit card company can identify and block risky transactions before they are closed.

Event streaming also can spark ideas and programs among line workers across the organization, helping re-shape their businesses. Farm equipment makers launch smart maintenance services that monitor vehicle sensors via satellite. Hedge funds devise new algorithmic stock trading programs. Strategies like this increase margins and create new revenue streams.

TCO reduction. Streaming architectures reduce CPU requirements and therefore cost by processing event data once, incrementally, rather than repeatedly batches of both new and old data. This also reduces the bandwidth required for data integration, which is particularly cost-effective when streaming data from an on-premises source over the Wide Area Network (WAN) to a cloud target.

Technical Benefits of a Streaming-First Architecture

Data teams can use event streaming to more easily execute projects and meet service-level requirements (SLAs). They can add data sources, absorb rising data volumes, and improve application response times without adding infrastructure.

Scalability. The efficiency advantage of incremental stream processing means that data teams can more easily accommodate the massive data sets required for machine learning and other types of AI. They also can use streaming to transform and filter large datasets before they arrive in a target platform, further increasing scalability. Data teams can compound these scale advantages by leveraging target cloud platforms with elastic resources.

Application performance. Real-time streaming reduces application response times, accelerating either operational or analytical workloads. For example, an insurance firm can synchronize transactional records at near-zero latency to process claims and rapidly assist customers. An online retailer can deploy a software robot to automatically confirm and deliver shipment status updates to a customer on demand.

Simplicity. New automated streaming tools help data teams reduce the burden on ETL developers and accelerate projects. These tools help less-technical users configure and monitor streaming jobs with little or no manual scripting, and with minimal risk of time-consuming human error.

Challenges

Organizations navigate numerous data management and architectural challenges that can grow more severe with a conversion to streaming-first architectures.

Strategic confusion. Business and data teams often incorrectly scope streaming use cases. Analysts and data scientists that treat real-time analytics as a “cure-all” might lose focus on other valuable insights. For example, real-time preventive maintenance should not replace historical troubleshooting and root-cause analytics.

Real-time transformations. Many current ETL or streaming solutions cannot perform the complex transformations of data in flight that today’s multi-sourced and high-scale analytics require. This forces data teams to hand-code their own stream processors, sucking up developer time, or accept higher latency by pushing transformation to the target. It also forces data teams to limit the scope of advanced initiatives such as machine learning.

Heterogeneity. Organizations continue to add architectural components that change data processes and ratchet up management burden. They add data sources, such as social media streams to gauge customer sentiment, IoT sensors to track machinery, or cloud-based Software as a Service (SaaS) to handle back-office tasks. They add targets, such as cloud data warehouses for structured data, data lakes for multi/unstructured data, or NoSQL for document stores. And they add custom and commercial software to transform and analyze the data. Each new component brings potential business value, but also more work.

Complexity. Data teams often struggle with a lack of automation. They execute cumbersome scripts for replicating, streaming and transforming data across their architectures. They often lack the necessary ETL or Apache open source development skills to deploy data pipelines on time. On-premises infrastructure requires vigilant monitoring and tuning to meet stringent SLAs.

Inflexible architectures. Traditional environments impede business-driven innovation. On-premises data warehouses are slow and costly to upgrade. Data teams struggle to deploy new components because they need custom coding to interoperate. They also struggle to unlock data from costly legacy systems such as the mainframe, which requires scarce scripting skills to convert outdated formats into something consumable.

Adoption Patterns

Market Trends

Organizations adopt streaming-first architectures as part of a broader evolution in their data strategy. Let's examine the most common adoption trends.

Data pipeline automation. As data teams build pipelines to connect various end points, they use automated tools that replace command-line scripting with a graphical interface. These tools enable architects to perform work themselves rather than waiting on busy developers. They create new streaming data pipelines faster, more easily and more cost-effectively.

Cloud-driven data modernization. Data teams offload analytics workloads from mainframe or other costly legacy systems to modern platforms. They migrate to cloud Infrastructure as a Service (IaaS) offerings, based on data lakes, data warehouses, or NoSQL, to address new use cases while streamlining costs. They develop software on cloud Platform as a Service (PaaS) offerings, and subscribe to Software as a Service (SaaS) offerings.

Smartphone Apps. Expectations on analytics have significantly shifted in the last ten years. Smart applications now rely on real-time data to deliver data intensive services to mobile consumers. They provide continuous updates to mobile news feeds, weather forecasts, crowdsourced traffic-monitoring services and other apps.

Machine learning (ML) Models. Enterprises build, train and deploy ML models, which learn from and adapt to data patterns without being explicitly programmed to do so. ML both enriches existing analytics initiatives and helps address new use cases. It requires high volumes of high-quality data to generate the most accurate results. This drives the need for efficient, real-time and/or low latency processing of high volumes of data.

Case Study

A *Fortune* 100 manufacturer of medical, automotive and power-generation equipment realized many of the streaming benefits outlined above by retiring its legacy batch-only data architecture.

This firm analyzes mountains of data to drive every aspect of its business, from product specification to design, production and delivery. Their global supply chain depends on a data warehouse that consolidates vast Enterprise Resource Planning (ERP) records and millions of IoT sensor signals. This EDW depended on a batch ETL process based on outdated Hadoop technology. MapReduce software transformed incoming data in batches across multiple staging areas, before it was loaded into the EDW for analysis. That legacy ETL approach choked on rapidly scaling data volumes, creating delays, driving up computing costs and tying up administrators.

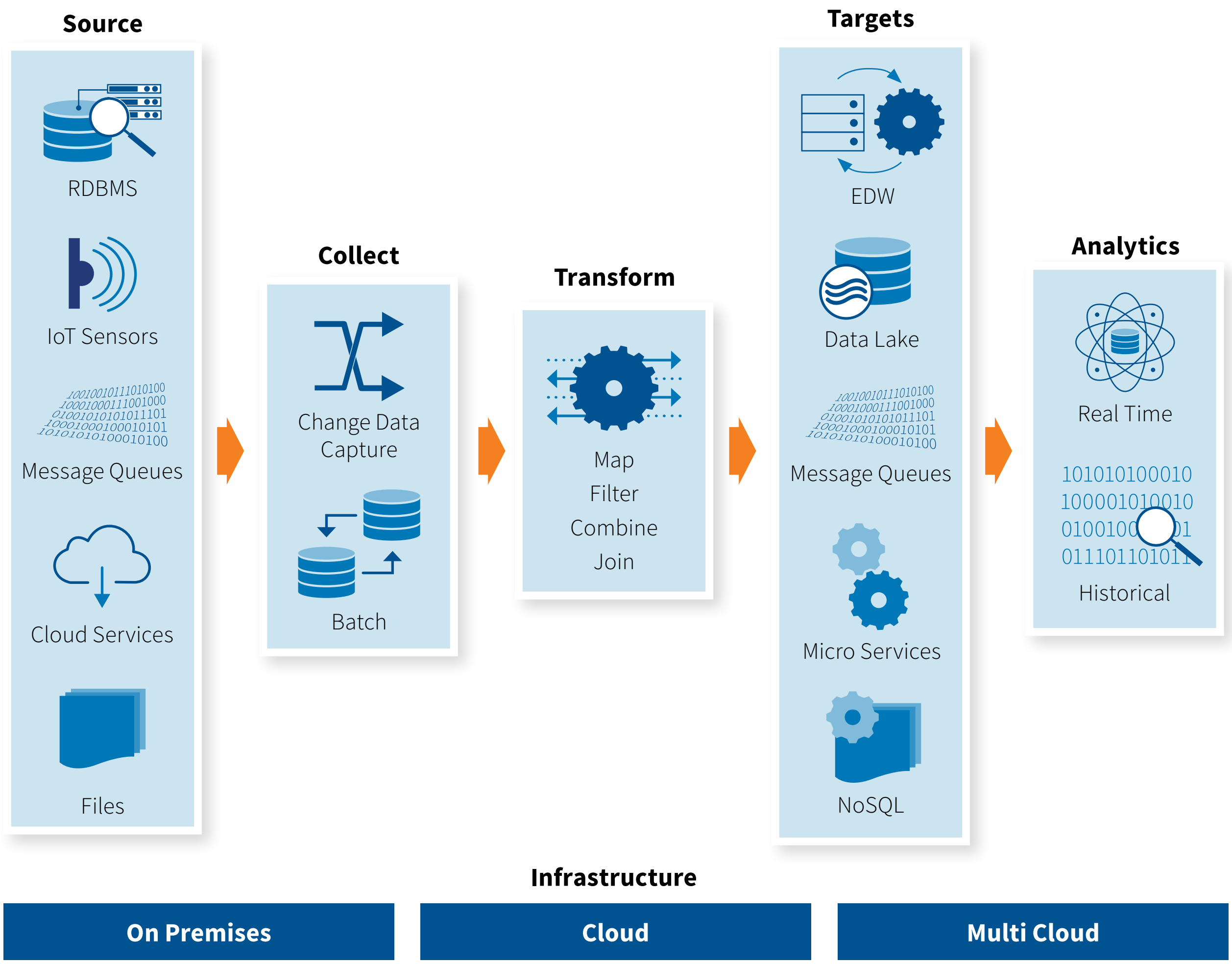
"We weren't able to set up or maintain data pipelines as nimbly as we needed to keep up with our customers," said their VP of Operations for the Americas. "And our total cost of ownership was spiraling out of control."

This Fortune 100 manufacturer decided to re-architect the system. Its data team converted to a new data pipeline to feed the data warehouse for continuous insights. They deployed a fully automated data ingestion tool that efficiently transforms event data in flight, reducing the processing steps 3x and improving performance 15x with commercial data ingestion technology based on Apache Spark. They now manage larger data flows than ever, consuming fewer CPU cycles at a lower cost. They cut development time 70%. The manufacturing firm next will expand the streaming pipeline to new parts of the supply chain, as well as back-office functions like billing.

Architecture

Data teams should design streaming-first data architectures with the following components. Typically they can incorporate these components into existing environments and migrate data and workloads to them incrementally, starting with higher-priority real-time use cases. Figure 3 illustrates a streaming-first reference architecture.

Figure 3. Streaming-First Reference Architecture



Data sources

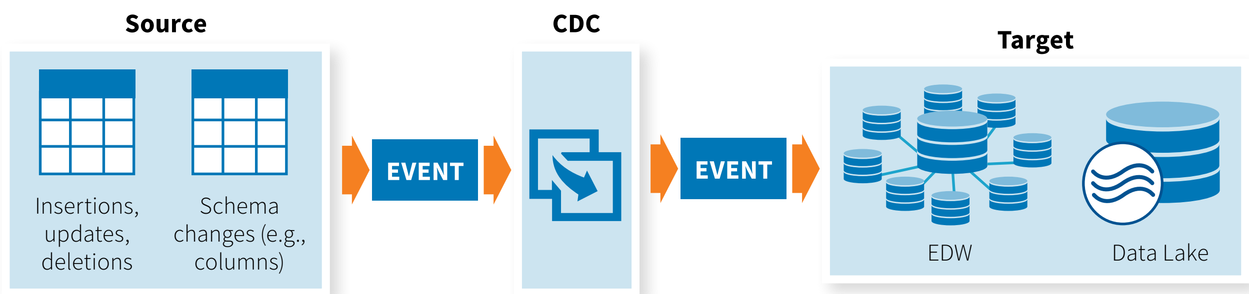
Organizations are awash in data. Database and mainframe applications process sales, supply chain, CRM, and other business records. Equipment sensors emit signals about operational performance. Websites and social media platforms track customer behavior. Organizations also procure more and more files from external data exchanges. For example, hedge funds might monitor weather pattern feeds to understand supply chain risks during hurricanes or floods. Business and data teams must take a surgical approach to scoping their data sources.

What data do their business clients really need? What people, tools and tasks are required to collect, stream and analyze that data? Answers to these questions help guide design choices for the next pieces of the architectural puzzle.

Data collection

Change Data Capture. Many organizations replace or complement legacy batch ETL with Change Data Capture (CDC) technology in order to break bottlenecks for high volumes of data coming from database sources. CDC identifies and copies events from a source in real time, for example to publish database record changes to an event stream. These record changes include inserted, updated or deleted rows. They also might include changes to source structures (a.k.a. schemas), such as a new or revised table column. Figure 4 illustrates CDC functionality.

Figure 4. Change Data Capture



Data teams can minimize disruptions to the performance of production applications by using CDC platforms that read backup logs rather than the production database. Such tools have a lower impact than those that query the database itself or track changes with extra “shadow” tables.

Batch collection. Organizations typically convert to streaming-first architectures in phases, targeting the most strategic and time-sensitive workloads first. They still collect (a.k.a. extract) some datasets in batch, provided the business can absorb some latency for those workloads, and provided the batch volumes do not strain available CPU capacity or bandwidth. Traditional batch scripting tools require developer expertise. Data teams can reduce this developer work by instead selecting modern automated solutions to handle both batch and streaming jobs.

Data Transformation

As with traditional systems, streaming-first architectures transform data in inter-related ways that grow more complex with the addition of each new data source or target. Open source software such as Apache Spark helps improve performance and enable more sophisticated tasks.

Commercial, open source and/or homegrown transformation software must map your data – that is, match source and target fields – to reconcile different models or formats. These technologies should transform and manipulate (e.g. aggregates, enrichments, and perform lookups on) source datasets. They should filter data helping users define and replicate only the subset needed at the target. Transformation and manipulation helps prepare data for real-time analytics and machine learning, which often focuses on specific fields within large datasets. Manual ETL scripts or more intuitive drag-and-drop GUIs perform lookups on historical datasets, then aggregate relevant records to enrich the data stream and make correlations. They also join tables or other subsets of data.

Organizations must decide where to perform these transformations: on data streams in flight, or on data after it comes to rest in the target? Target platforms often have more processing power and can therefore transform data more efficiently on data after it comes to rest. But this comes at a high cost of latency. Certain streaming tools can execute the necessary transformations on event data in flight at low latency, thereby enabling real-time insights and actions. When the lowest-latency insights are required, data teams should seek streaming architectures and tools that perform the required transformations with minimal performance impacts. New and automated streaming tools, particularly those with robust transformation capabilities and advanced graphical user interfaces, can meet requirements and simplify streaming processing.

Targets

Organizations typically divide analytics workloads among multiple platforms. For years they ran daily, weekly or monthly operational reports on on-premises data warehouses. Recently data teams migrated some of those workloads to cloud-based data warehouses that replace onsite infrastructure with elastic, pay-by-the-drip compute and storage. Cloud data warehouses support visual dashboards, reports, and ad hoc queries more flexibly and cost-effectively than their on-premises predecessors.

Organizations also moved their advanced analytics workloads from on-premises Hadoop data lakes, to cloud-native, pre-packaged data lakes or NoSQL data stores that simplify the collection and analysis of multi-, semi- and unstructured data. Data scientists run many of their most advanced analytics workloads – for example, simulations, predictions and recommendations – on cloud data lakes that can flexibly store and process the necessary bots, machine learning models, etc.

Event streaming makes both data warehouses and data lakes more efficient and scalable. They can ingest event streams continuously and incrementally, which reduces latency and frees up resources to handle more data. As they design their architectures, data teams should consider their latency tolerance and processing requirements. The more transformation they can perform on the data in flight, the less they will need to run on data once it lands in the data warehouse or data lake.

Data teams also publish event streams to Kafka or other messaging systems that further process data in flight for real-time analytics and operations. Kafka and recent vendor-driven variants such as Azure Event Hub move data at higher scale, higher throughput, and lower latency than predecessors such as IBM MQ. Kafka's intermediate brokers enable microservices to share data while remaining independent and autonomous. Given their focus on continuous incremental processing, both messaging and microservices targets integrate naturally with event streaming as a source.

Analytics

Real Time. Analysts configure Business Intelligence (BI) and data science tools to automatically correlate events and windows, draw historical comparisons, and apply scoring methods to understand implications. Their rules and models automatically measure impacts, predict outcomes, and recommend actions. For example, a smartphone application can check location and purchase history patterns, then recommend special offers to a shopper in a grocery store.

Streaming analysis might include Artificial Intelligence (AI), which applies computing logic to tasks that normally require human cognition. A common component of AI is machine learning (ML) software, which teaches itself data patterns after being "trained" to do so on historical data. Data teams can apply ML models to streaming data, then adjust them based on results, either periodically or automatically on an ongoing basis.

Historical. While the term might strike a negative tone in today's time-sensitive business environment, historical analytics remain critical to both operational decision making and innovation. Executives need to study and compare financial performance across quarters. Managers need to measure trends in inventory levels over time at each point in the supply chain. Innovation itself usually relies on the past. Data scientists build their most valuable insights predictions by drilling into large and varied data sets that can only be assembled over longer periods of time.

Infrastructure

Most organizations maintain a mix of on-premises, cloud or even multi-cloud infrastructures. Enterprises typically rely at least in part on traditional ERP, CRM or other business applications that run on database or mainframe systems in their data centers. They modernize by offloading queries to modern analytics platforms like data lakes or new data warehouses on the cloud. They also incrementally migrate operational tasks to cloud-based applications.

Infrastructure as a Service (IaaS) offerings from Cloud Service Providers (CSPs) such as AWS, Azure or Google Cloud Platform absorb more and more of these workloads. They help organizations eliminate a lot of IT management work, and smooth out costs by converting capital expenditures to consumption-based operating expenditures. Data and analytics teams also tackle new use cases by testing, customizing and building advanced analytics tools on these CSP platforms. As they learn, they often branch to a second or even a third CSP whose specialty best fits their needs.

Your Journey to a Streaming-First Architecture

So where to start? Data teams struggle just to maintain the status quo, keep up with rising data volumes, and prioritize scarce IT resources to meet ongoing business requests. Your journey to a streaming-first architecture can pose overwhelming choices. Business and data teams must keep things simple where possible as they design and implement streaming-first architectures in these sprawling environments. Here are some essential guidelines.

1. **Start by alleviating pain.** Most data teams can easily identify one or more bottlenecks, where batch processing cannot keep up with rising data volumes, and the business is suffering as a result. Eliminate your greatest choke point by converting legacy ETL – usually the culprit – to CDC or possibly Kafka streaming. Use a data integration tool that automates stream configuration, execution and monitoring to free up programmer time and reduce the risk of error. Keep things simple with this first project, focusing on just a few end points that can be integrated without custom work.

   Stopping the pain in this way clears room for higher data volumes and velocities. It eliminates delays and creates new real-time possibilities for business data consumers. It also establishes a quick win that you can use to gain funding for additional projects.

2. **Connect additional sources and targets to your streaming architecture.** Once you get a quick win, more business users will ask to get in on the game. Fan out your CDC and Kafka streams to absorb more end points, retiring more ETL processes along the way. Consider using a data integration tool or Kafka APIs that easily connect to all your sources and targets. Also consider data integration tools that can scale to handle complex transformations of data in flight without incurring significant latency.

3. **Proceed to more advanced use cases.** With a streaming-first architecture now well established, you can start to scope more innovative real-time analytics projects. These might include proactive customer engagement, IoT operations, or even robotic process automation. Carefully scope the skills, resources and success factors of complex projects before committing to them.

4. **Embrace cloud Infrastructure as a Service (IaaS).** Take advantage of IaaS offerings from CSPs in order to minimize IT administrative work, gain financial flexibility, and increase scalability with elastic resources. All major IaaS platforms now include data lake, NoSQL, data warehouse and streaming options to support a wide range of operational or analytics use cases.

5. **Minimize lock-in risk.** Stay focused on the need for data mobility as you test new cloud platforms and tools to address new use cases. Approach Platform

as a Service (PaaS) cautiously, because many PaaS tools have platform-specific configuration requirements that make future data migration tricky. As your team learns and business requirements evolve, your datasets need to maintain unfettered access to new platforms and tools wherever possible.

In a cost-conscious business environment, streaming-first data architectures offer a rare opportunity for quick payoff. If you start by dislodging some of those batch bottlenecks, you can create technical and budgetary headroom to embark on a more ambitious journey to real-time, advanced analytics. Start learning about these near term and long-term opportunities today.

About Eckerson Group

Wayne Eckerson, a globally-known author, speaker, and advisor, formed Eckerson Group to help organizations get more value from data and analytics. His goal is to provide organizations with expert guidance during every step of their data journey. Today, Eckerson Group helps organizations in three ways:

- **Our thought leaders** publish practical, compelling content that keeps you abreast of the latest trends, techniques, and tools in the data analytics field.
- **Our consultants** listen carefully, think deeply, and craft tailored solutions that translate your business requirements into compelling strategies and solutions.
- **Our educators** share best practices in consulting workshops or external conferences on 30+ topics that can be tailored to your needs.

Our experts each have more than 25+ years of experience in the field. They specialize in data analytics—from data architecture and data governance to business intelligence and artificial intelligence. Their primary mission is to help you get more value from data and analytics by using their extensive experience.

Our clients say we are hard-working, insightful, and humble. It all stems from our love of data and our desire to help you get more value from analytics—we see ourselves as a family of continuous learners, interpreting the world of data and analytics for our clients and partners.

Get more value from your data. Put an expert on your side.

Learn what Eckerson Group can do for you!

About Equalum

Business leaders in data-intensive organizations are increasingly expected to unify fragmented data from across silos to power decision making in real time. But traditional changed data capture (CDC) tools and legacy ETL processes are straining under a constantly-growing volume and velocity of data. And in-house implementations of powerful open-source frameworks are complex and costly to implement and maintain. Information technology leaders need a new approach.

Equalum is the fastest data ingestion platform, relied upon by enterprises across industries to seamlessly stream data to operational, real-time analytics, and machine learning environments. Built for scalability and ease of use, Equalum ingests data in real time, as it is created, from any number of data sources. It processes and transforms the data before streaming it to any number of target applications or systems. Equalum technology harnesses the power of Apache Spark and Kafka, among other cutting-edge open source technologies, helping organizations rapidly accelerate past traditional CDC, ETL, or open-source implementations with a zero-coding approach, intuitive design, and minimal maintenance.

The Equalum data ingestion (DI) platform provides:

Enterprise Grade DI Manage all your data pipelines in a single data ingestion platform.	**Infinite Speed and Scalability** Linearly scale your data pipelines to handle any volume and speed of data.	**Self Service & Zero-Coding** Develop and deploy all your pipelines from an easy-to-use, single-pane interface.
Streaming and Batch Data Processing Support all your data ingestion use cases with powerful CDC and replication capabilities.	**Robust Data Transformations and Manipulations** Go beyond CDC and data replication with sophisticated ETL-like capabilities.	**Seamless Ecosystem Integration with Minimum Overhead** Connect to all your data sources and targets with minimum operational impact.
Comprehensive Support for OSS Data Frameworks Leverage the power of Spark and Kafka in a risk-free environment.	**Flexible Deployment with a Platform Agnostic Architecture** Run your data pipelines anywhere you need them.	**Autonomous Data Exploration** Radically increase productivity and minimize errors with automated data ingestion.

Learn more about Equalum or schedule a demo today.