Data Processing in Modern Businesses

ETL
ELT
ETLTL...
ELTLT...
Streaming
Real-time Processing
Event-Driven Data Processing

Oli Kelly | 13' Feb 2025

Modern businesses process data in many different ways.

Choosing the right method impacts performance, cost, and compliance.

Recap: ETL
Extract Transform Load

🌱 First appearance

1970s. Introduced alongside early relational databases (IBM, Oracle)

⚙️ What does it do?

Extracts data from source, transforms it into a structured format according to requirements, and loads that structured data into a target database for analysis and/or storage.

💡 What problem does it solve?

Ensures data quality and structure before storage, making it easier for businesses that require compliance and clean, structured data. Ensures data quality and compliance before ingestion.

🤝 Where is it commonly used?

Banking (fraud detection), healthcare (patient records), airlines (scheduling).

JP Morgan, NHS, Delta Airlines.

😱 So what's the catch?

Slow ingestion, less flexible for schema changes.

Handling large batch loads, ensuring transformation consistency, managing schema updates.

ELT
Extract Load Transform

🌱 **First appearance**

2000s – Emerged with cloud data warehouses like Snowflake, BigQuery, and Redshift, enabling scalable raw data storage.

⚙️ **What does it do?**

Extracts and loads raw data first, then transforms it within the data warehouse.

💡 **What problem does it solve?**

Solves the issue of flexibility in querying and analytics, allowing raw data to be stored and transformed later based on evolving needs.

🤝 **Where is it commonly used?**

E-commerce recommendations, media content tracking, cloud analytics.

Amazon, Netflix, Spotify.

😱 **So what's the catch?**

Higher storage costs, requires strong warehouse compute power.

Tech challenges: optimizing query performance, managing compute costs, ensuring transformation consistency.

ETLTL...
Loop

🌱 **First appearance**

1990s – Financial institutions and AI research began iterative transformation loops for fraud detection and model tuning.

⚙️ **What does it do?**

Extracts, transforms, loads, then loops back for further transformation as data evolves.

💡 **What problem does it solve?**

Useful for evolving datasets that need reprocessing. Addresses the need for continuous data reprocessing and refinement, useful for fraud detection and AI model updates that require iterative learning.

🤝 **Where is it commonly used?**

Fraud detection (iterative scoring), AI model training, recommendation system tuning.

Visa, OpenAI, TikTok.

😱 **So what's the catch?**

Computationally expensive, requires continuous updates.

More technically complex to implement and maintain. Versioning transformed data, managing iterative changes, handling real-time updates.

ELTLT...
Loop

🌱 First appearance

2010s – Growth of AI and machine learning led to dynamic transformation loops for constantly updating models.

⚙️ What does it do?

Extracts, loads raw data, then applies multiple rounds of transformations dynamically.

💡 What problem does it solve?

Solves the problem of evolving datasets where transformations need to be applied dynamically at different stages for different use cases.

🤝 Where is it commonly used?

AI-driven chatbots, financial risk models, dynamic ad targeting. Great for AI, machine learning, and event-driven analytics.

ChatGPT, Google Ads, Bloomberg.

😱 So what's the catch?

High storage costs, complex orchestration.

Maintaining consistent data transformations, ensuring backward compatibility, managing computational overhead.

Event-Driving Data Processing
Triggered

🌱 First appearance

2010s – Rise of microservices and cloud computing (AWS Lambda, Kafka) enabled event-driven architectures to replace inefficient polling methods.

⚙️ What does it do?

Triggers actions based on events rather than scheduled processing.

💡 What problem does it solve?

Enables automation and efficient resource use. Solves the inefficiency of traditional polling-based architectures by enabling automated, event-driven workflows that process data only when needed.

🤝 Where is it commonly used?

E-commerce order updates, serverless applications, IoT devices. Amazon Web Services (AWS Lambda), Uber, Ring (smart home security).

😱 So what's the catch?

Difficult debugging, event failure handling is pretty complex. Ensuring event delivery guarantees, managing idempotency, handling event retries and failures.

.

Streaming
Stay Ready

🌱 **First appearance**

2010s – Big Data movement (Hadoop, Spark) introduced real-time ingestion to replace slow batch processing.

⚙️ **What does it do?**

Processes data continuously as it arrives, without waiting for batch intervals.

💡 **What problem does it solve?**

Low-latency insights and real-time analytics. Solves the problem of real-time data ingestion for analytics, allowing organizations to gain insights instantly rather than waiting for batch processing.

🤝 **Where is it commonly used?**

Stock market updates, live sports analytics, cybersecurity alerts. NYSE, ESPN, Cloudflare.

😱 **So what's the catch?**

Difficult to maintain stateful data, requires robust infrastructure. V complex; Ensuring exactly-once processing, managing out-of-order events, optimizing real-time performance.

Real-Time Processing (RTP)
Stay Ready

🌱 **First appearance**

2015+ – Demand for ultra-low-latency processing in self-driving cars, IoT, and finance led to near-instant decision-making systems.

⚙️ **What does it do?**

Processes individual data points immediately upon arrival, typically for mission-critical tasks.

💡 **What problem does it solve?**

Ultra-low latency (faster than streaming). Provides instant responses to high-priority events. Solves the need for fast decision-making, where mission-critical applications require immediate processing and response.

🤝 **Where is it commonly used?**

Self-driving cars, emergency alerts, fraud prevention. Tesla, FEMA, Mastercard.

😱 **So what's the catch?**

Expensive infrastructure, requires high fault tolerance. V complex;. Ensuring ultra-low latency, handling system failures, scaling under high traffic.

Data processing at NETFLIX
A Case Study

Data Type	Processing Method	Why this method?	Example feature
User Viewing Data	ELT	Stores raw viewing data to allow flexible, on-demand transformations for personalized recommendations and content insights.	Personalized recommendations ('Because you watched…')
Real-Time Streaming Metrics	Streaming	Monitors streaming performance in real-time to ensure optimal user experience and immediate issue resolution.	Adaptive bitrate adjustments (video quality changes dynamically)
A/B Testing Results	Batch Processing	Aggregates and analyzes experimental data to inform decisions on new features and interface changes.	Testing out new UI features.
Error Logs and System Events	Event-Driven Processing	Triggers alerts and automated responses to system anomalies, maintaining service reliability.	Auto-scaling servers during periods of peak traffic.

The Bottom Line
What you need to know

ETL → Best for compliance-heavy, structured data.

ELT → Prioritizes speed and flexibility, storing raw data first and transforming it later in the cloud.

ETLTL → Used when AI needs constant retraining based on new patterns and insights

ELTLT → Best for AI systems that modify their own data processing logic (eg. dynamic pricing, algorithmic bidding).

Event-Driven Processing → Best for automating workflows efficiently.

Streaming → Best for live monitoring and continuous data updates.

Real-Time Processing → Best for mission-critical instant decisions.

🙇🏼

Ta
Cheers
Thanks
for listening

Data Processing in Modern Businesses

Oli Kelly | 13' Feb 2025