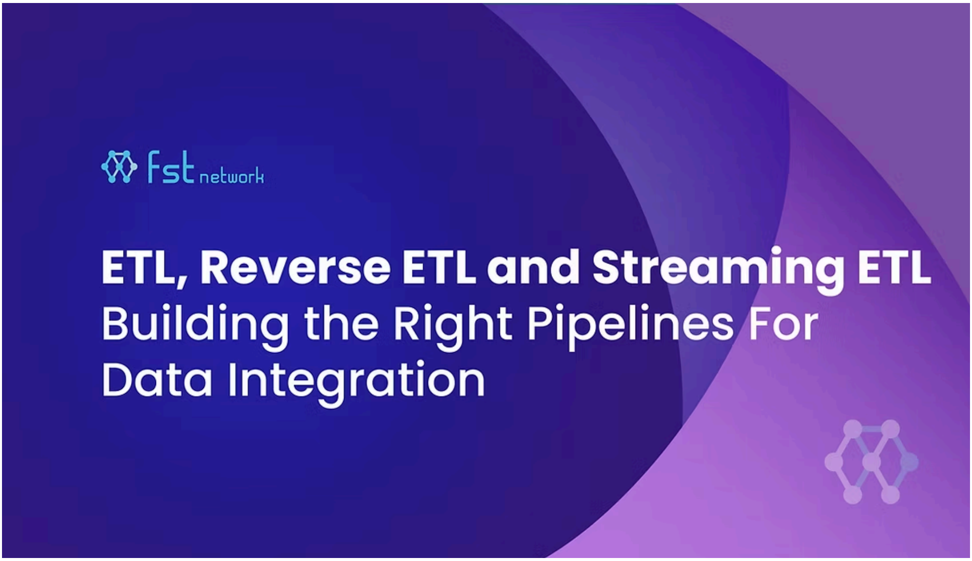
Nov 28, 2022 · 4 min read

ETL, Reverse ETL and Streaming ETL: Building the Right Pipelines For Data Integration

Updated: Dec 9, 2022

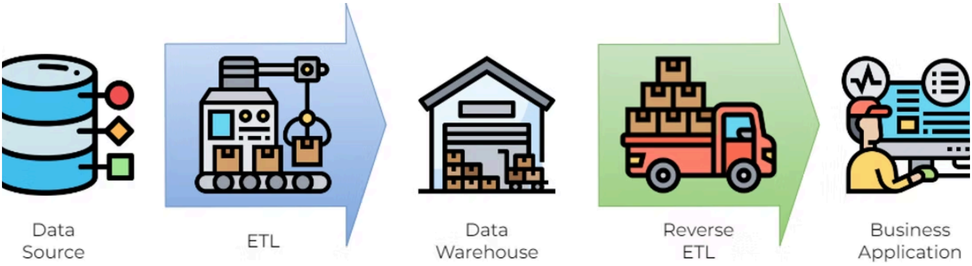

fst network
ETL, Reverse ETL and Streaming ETL
Building the Right Pipelines For Data Integration

What is ETL (and Reverse ETL)?

ETL stands for extract, transform and load, which is the traditional data integration approach emerged with data warehouses back in the 1970s.

In the middle of the 20th century, the rapidly-growing and increasingly accessible computers made people realised that it's possible to have the computer collect and analysis data to support decision-making. And the invention of time-sharing in 1960s makes these computers - still as large as a small room back then - can be used by multiple users at the same time. The idea of Decision Support Systems (DSS) and Business Intelligence (BI) was born around that time, as well as the first database management system developed by IBM.

Fast forward to 1980s, the need of handling more data and eliminating duplication across databases gave birth to data warehousing, which is to collect current and historical operational data, transform them into analyzable data, then store in a big central database. The data would then be fed into BI systems to generate analytical reports, which may contain valuable insights for decision-making.



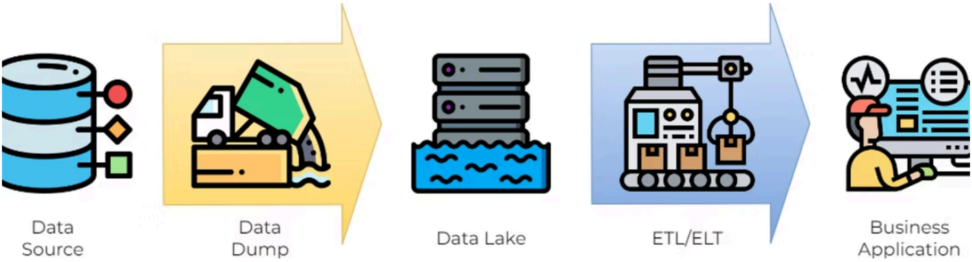
The process of extracting and processing data from sources to data warehouses are the so-called ETLs. They are usually scheduled batch jobs that run automatically. Today databases, ETLs and data warehouses are often seen as an essential part of BI. ETLs can be implemented in any way, but SQL script is one of the most commonplace forms.

Reverse ETL is similar to ETL except the source and target is reversed - to extract and transform data *from the data warehouse to an application*, since specific analytical data can also be useful for the right users at the right time and right place. This practice is called analytics operationalisation. Combined with data mining, data science or machine learning, you can apply insights directly to where they are truly needed - for example, customer relations or sales system.

ETL, ELT and Data Lake

As the name suggests, ETL transforms data *after* extraction, and the transformation would happen in a staging storage outside the data warehouse. ELT (extract, load, transform) does it a bit differently: the transformation happens directly in the data sotrage. This characteristic is one of reason that ELTs are often associated with data lakes, which started to appear around 2010s.

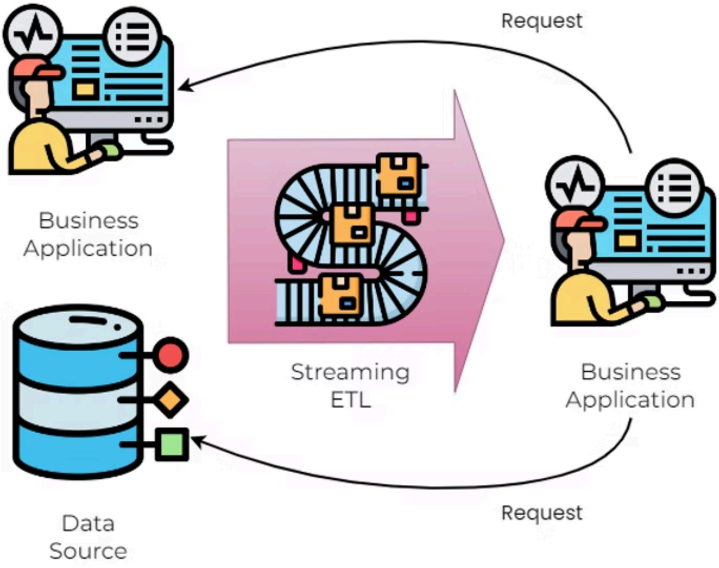
A data lake stores *unstructured* data, which does not require transformation in the first place. Big data - which are growing exponentially (2 zettabytes in 2010 and estimated 181 in 2025), even more so in the age of machine learning - can thus be “dumped“ in a much quicker and cheaper manner. The data still need processing to be used, of course, and these pipelines are also referred as ETLs or ELTs.



The problem of data lakes is that they *do not* guarantee data can be useful, and ETLs (or ELTs) may become bottlenecks just like those in data warehouses. Too often they'll just become data swamps, or worse, data *graveyards*.

Streaming ETL or Data Streaming: Event-Driven Pipelines

Like what we've mentioned in the data mesh and data product posts, the recent trend is to break down the centralised data model and productised the data as decentralised pipelines or data services. Thanks to the development in containers and Kubernetes, this is a lot easier to be done then before, and data products will now exist on a unified virtual plane (the goal of data virtualisation or data fabric) despite the actual storages and infrastructures may be distributed.



Does this mean we no longer need a data warehouse or a data lake? Maybe not - it really depends on your needs and use cases. Some companies, like Databricks, are focusing on an upgraded data lake model called **data lakehouse**. Some, like Snowflake, provides virtual data warehouses built upon AWS, Google and Azure cloud services. But this also means the ETLs can now deliver data without having to go through a central hub, which may eliminate the traditional bottlenecks as well as data duplication issues.

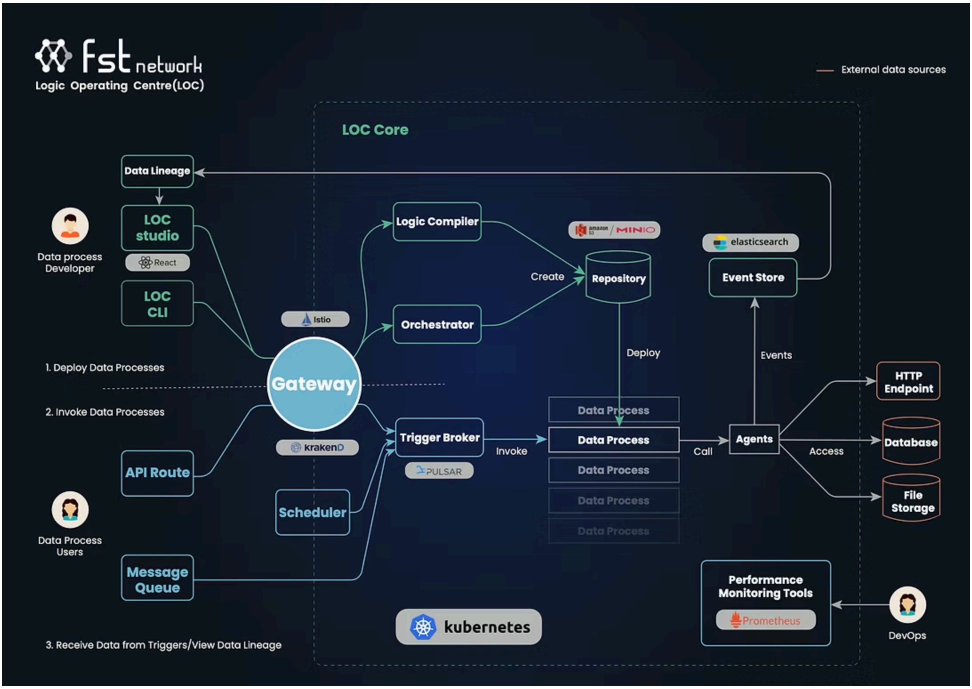
These new ETLs also shift from batch jobs to event-driven architecture (EDA). Instead of moving data periodically, a pipeline *only runs when* the target call for the data. This is the so-called **streaming ETLs**, or a fancier term, **data streaming**. Data streaming pipelines are often in the form of microservices or utilising message queue tools, for example, Apache Kafka.

Pipelines Can You Build with LOC

FST Network's Logic Operating Centre (LOC), despite deeply influenced by the data mesh model, does not limit itself in terms of data integration. You can actually use it in anyway you prefer, integrating the pipelines with your existing systems and data sources.

LOC offers three different types of *triggers* which can be used to invoke data processes: **API Routes** (HTTP endpoints)**, message queues,** and **schedulers**.

Trigger Type	Equivalent to (when combined with data processes)
Schedulers	(Batch job) ETLs/ELTs/reverse ETLs
API routes (synchronous)	Event-driven microservices or FaaS functions
API routes (asynchronous)	Streaming ETLs
Message queues	Streaming ETLs



You can see that LOC data processes are surprisingly versatile - the behavior *changes* depending on the triggers. All trigger types can be deployed and removed by users at runtime without having to upload complicated Kubernetes resource definitions. This also makes it possible to create hybrid ETLs that can do different things at different time.

Under the hood, LOC utilise a Apache Pulsar MQ service to convert all triggers to messages and broadcast them to LOC's internal data process runtime. An executed data process is referred as a **task**. Once a task is completed, the result would be returned directly to the invoker (synchronous) or stored away to be queried later (asynchronous).

From each data process you can also operate various data sources using LOC **agents** - the internal session storage, the data event store, as well as external HTTP endpoints, databases and file servers. In other words, these pipelines can have *multiple* input and output with different transformation between. Building data pipelines has never been this easy and yet flexible.

The data processes are extremely easy to deploy too, thanks to LOC's FaaS (Function as a Service)-like deployment design. We will talk about how exactly is it like, and what does it have to do with Kubernetes in one of the future articles.

FST Network aims to revolutionise the data landscape by creating world's first cloud-native and serverless Data Mesh platform. Logic Operating Centre (LOC) is fully capable to productise and eventify your data, with easy-to-deploy data pipelines that seamlessly connects everything across your organisation, bringing in agile, accessible and trustworthy business insights. The data race is constantly changing and fiercer than ever - but we are there to give you a winning edge.

Icon credits: flaticon.com