

**SFIT**

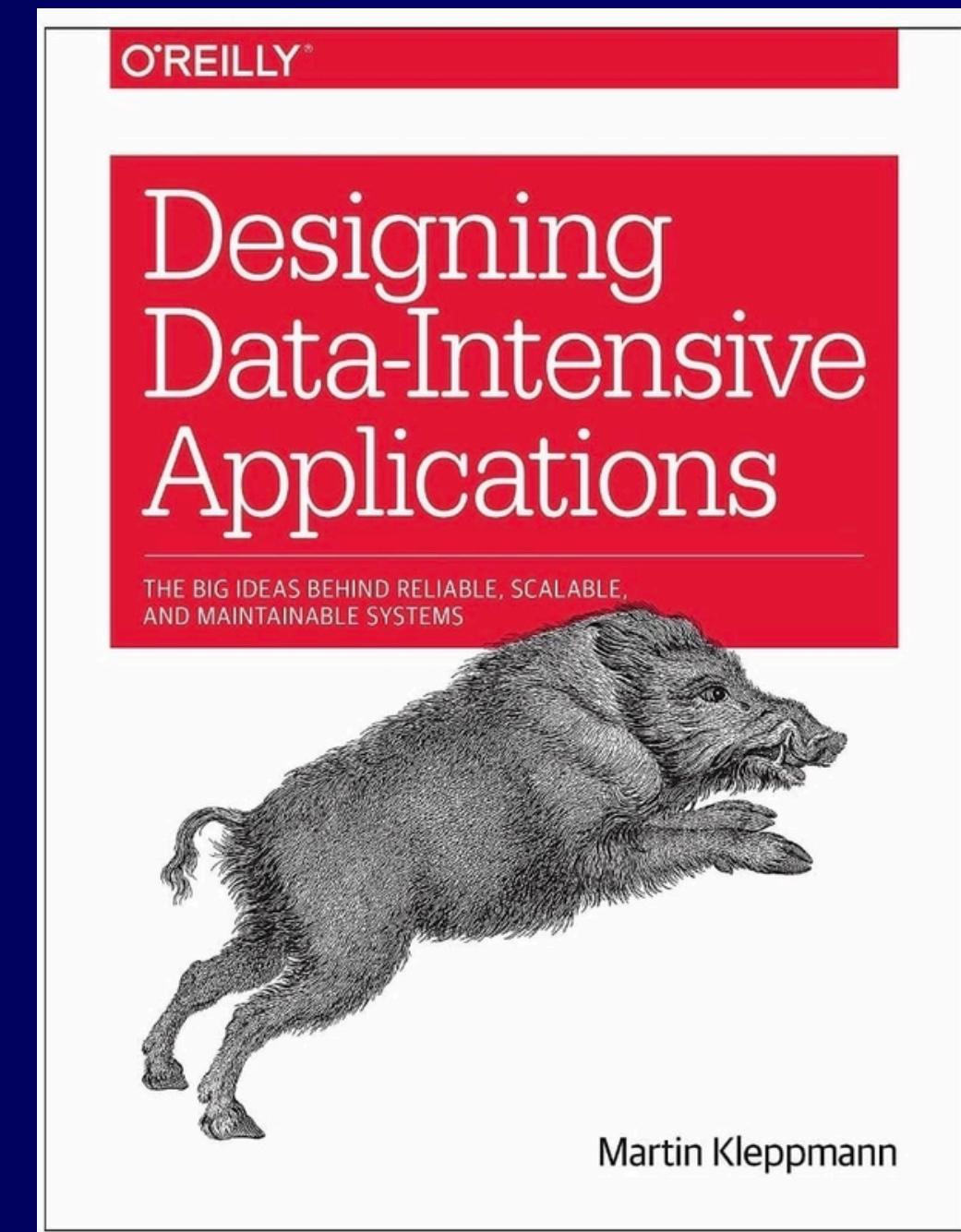
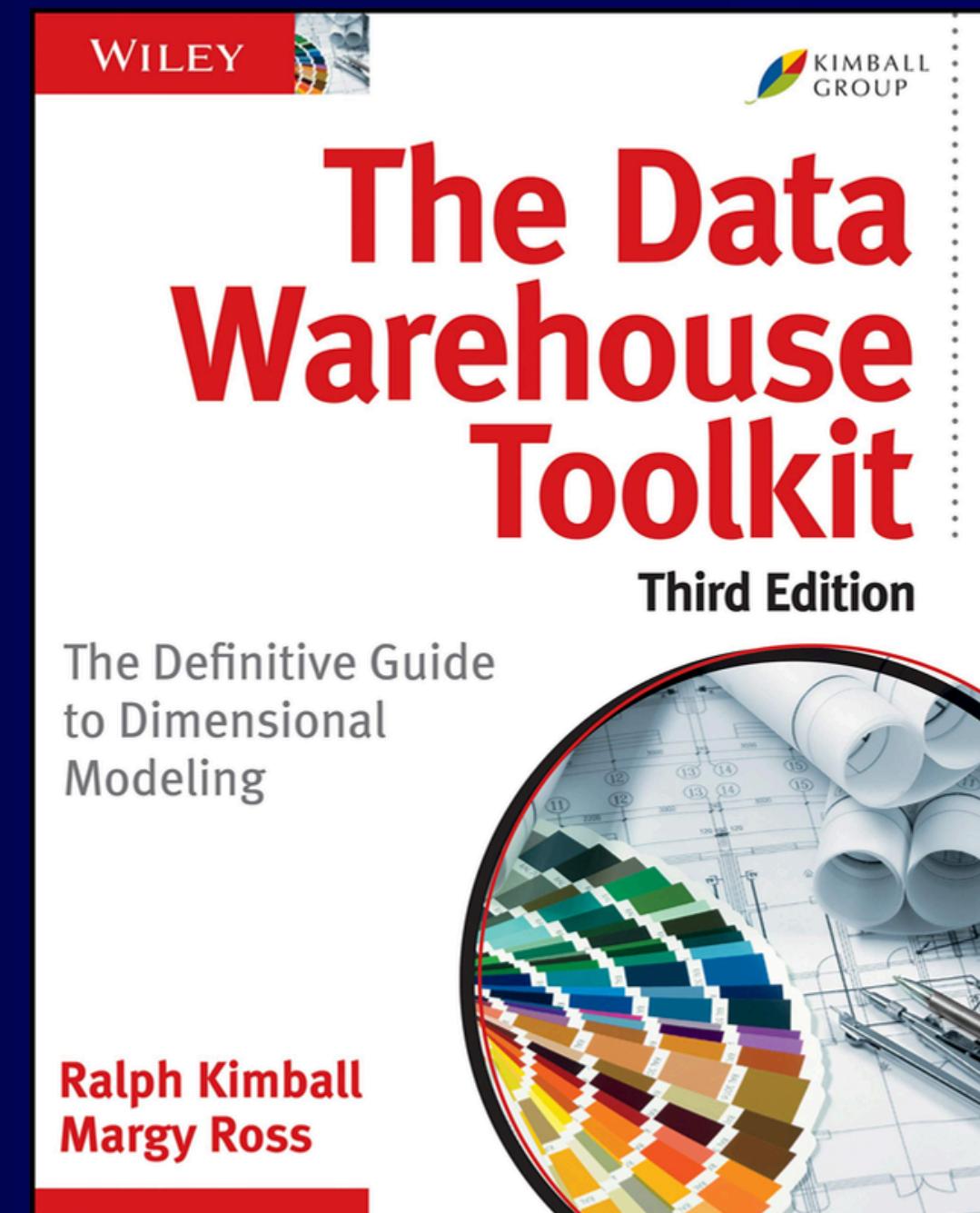
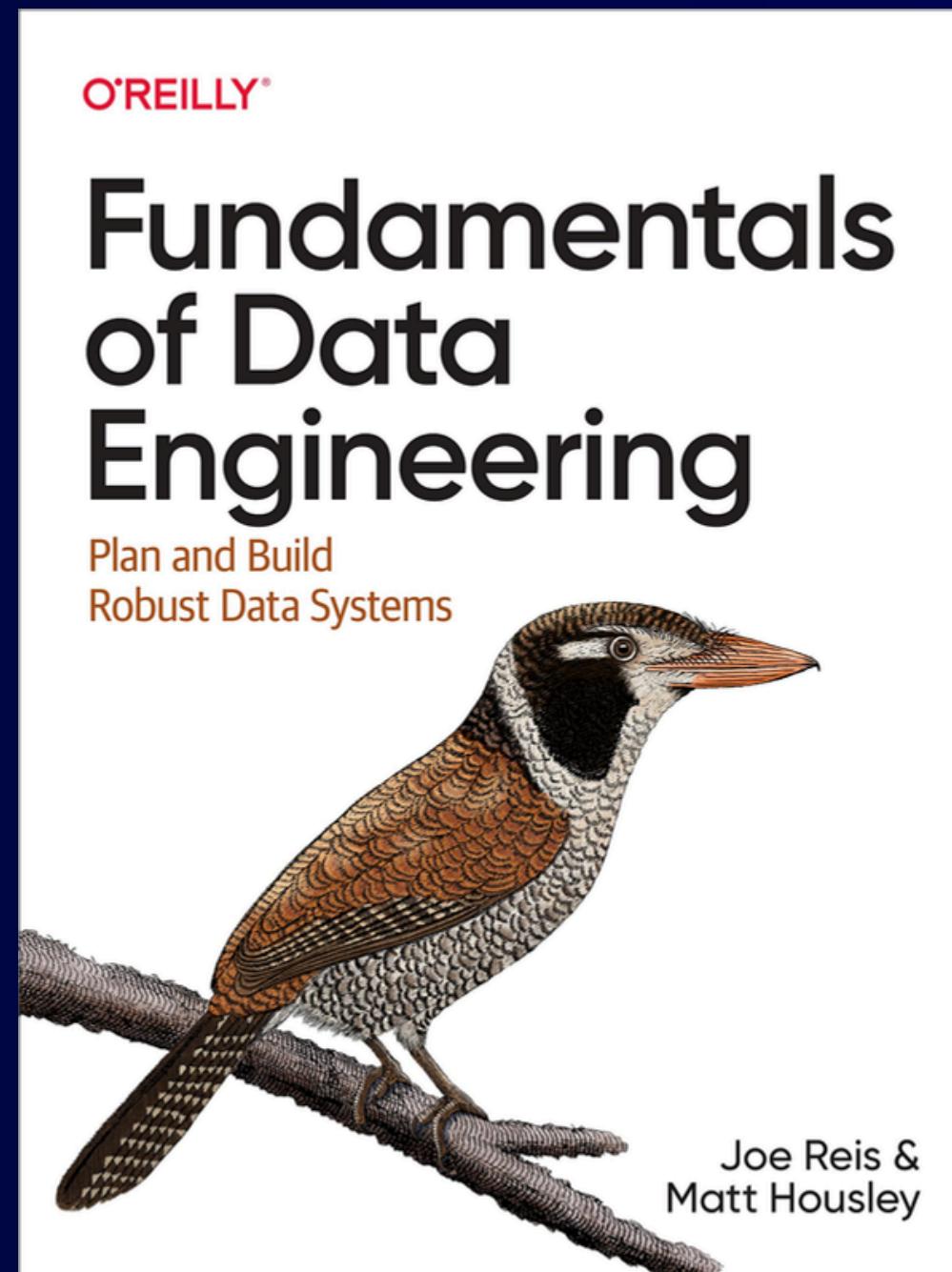
Trường đại học Giao Thông Vận Tải

**DATA  
ENGINEER**

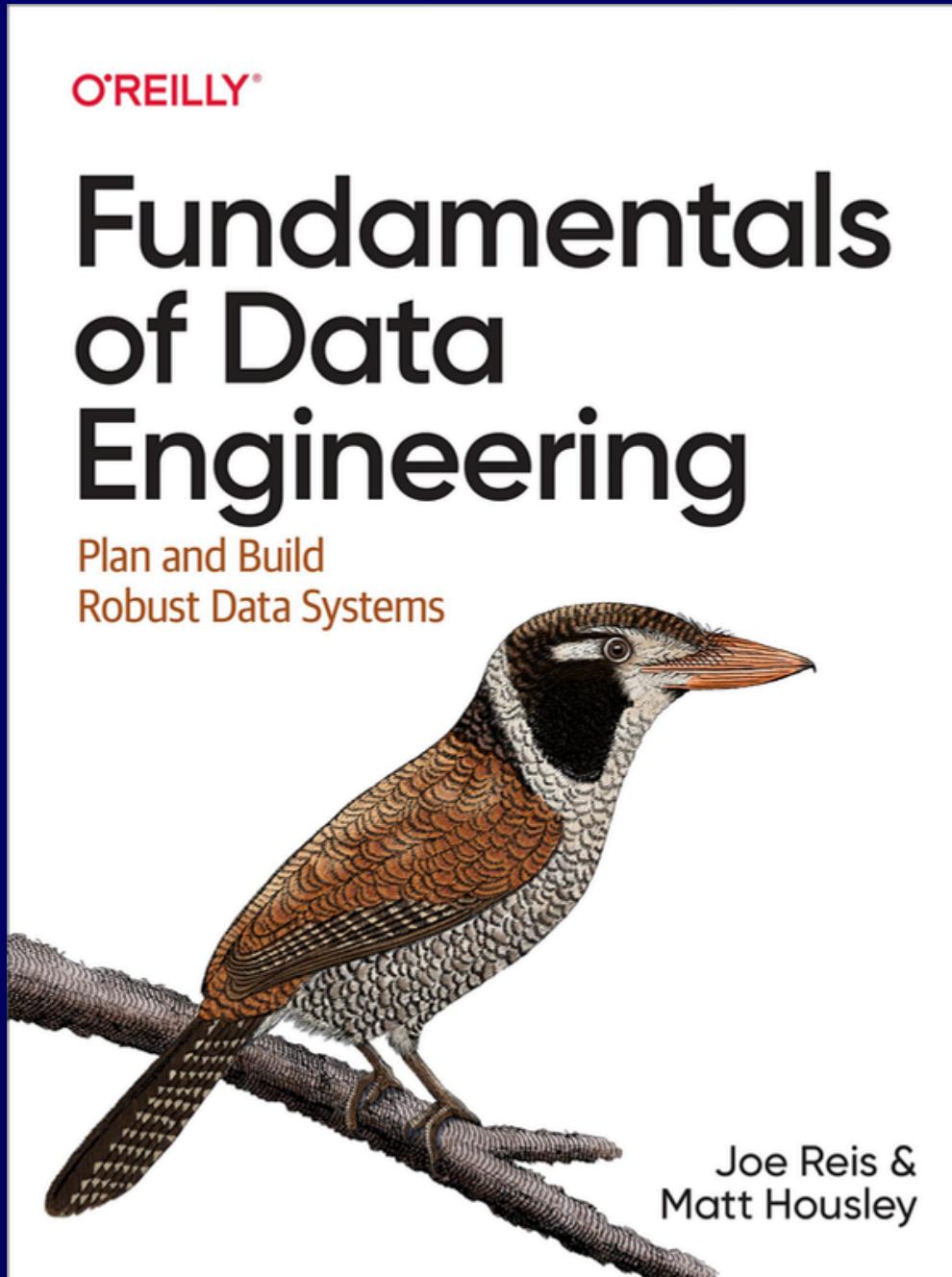
# Yêu cầu khi tham gia

- Hiểu biết cơ bản về ngôn ngữ lập trình C++, Java, Python... (Đặc biệt là Python và các thư viện như Pandas, Numpy...).
- Biết cơ bản về SQL và cách sử dụng những lệnh SQL cơ bản.
- Hiểu biết cơ bản về các thành phần trong Software Engineer

# Sách hay cho Data Engineer

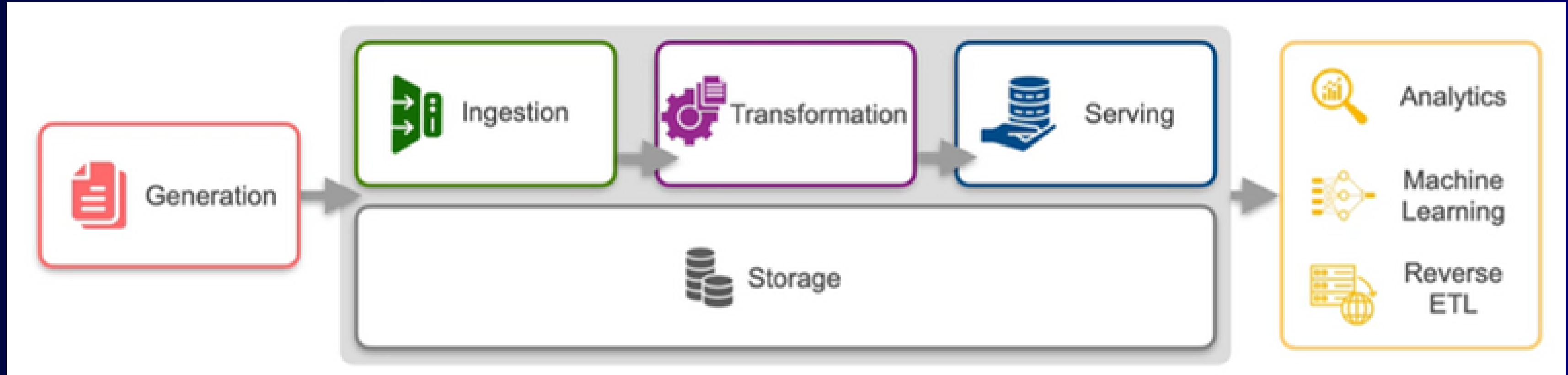


# Data Engineering là gì?



“Data Engineering là việc phát triển, triển khai và duy trì những hệ thống và các quy trình xử lý dữ liệu thô và cung cấp những thông tin chất lượng cao và thích hợp để hỗ trợ cho những công việc đằng sau, như phân tích dữ liệu hay machine learning (học máy). Data Engineering là sự giao nhau giữa bảo mật, quản lý dữ liệu, DataOps, kiến trúc dữ liệu, điều phối và kỹ thuật phần mềm.”

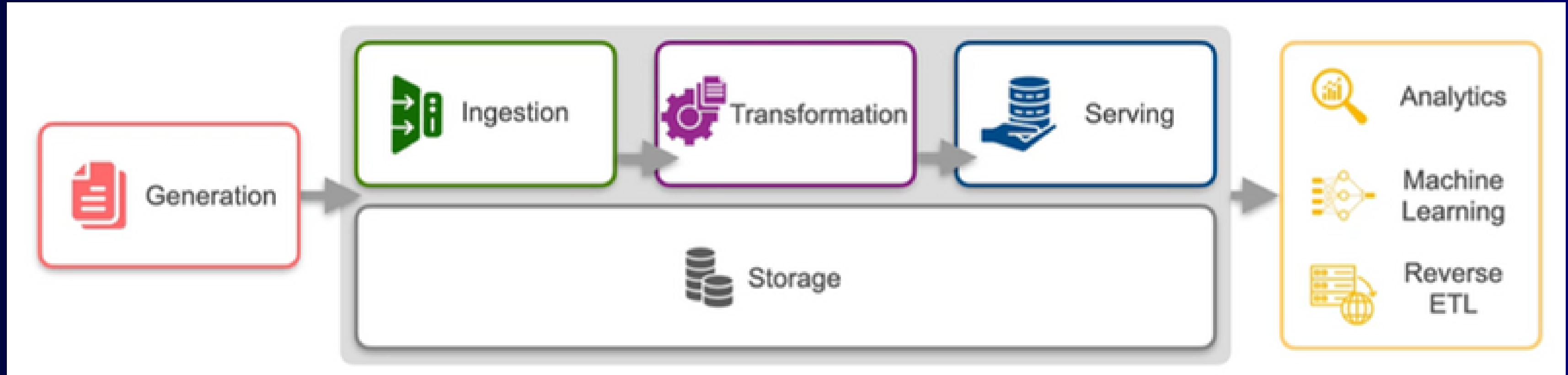
# Vòng đời của một Data Engineer



**Data Pipeline**

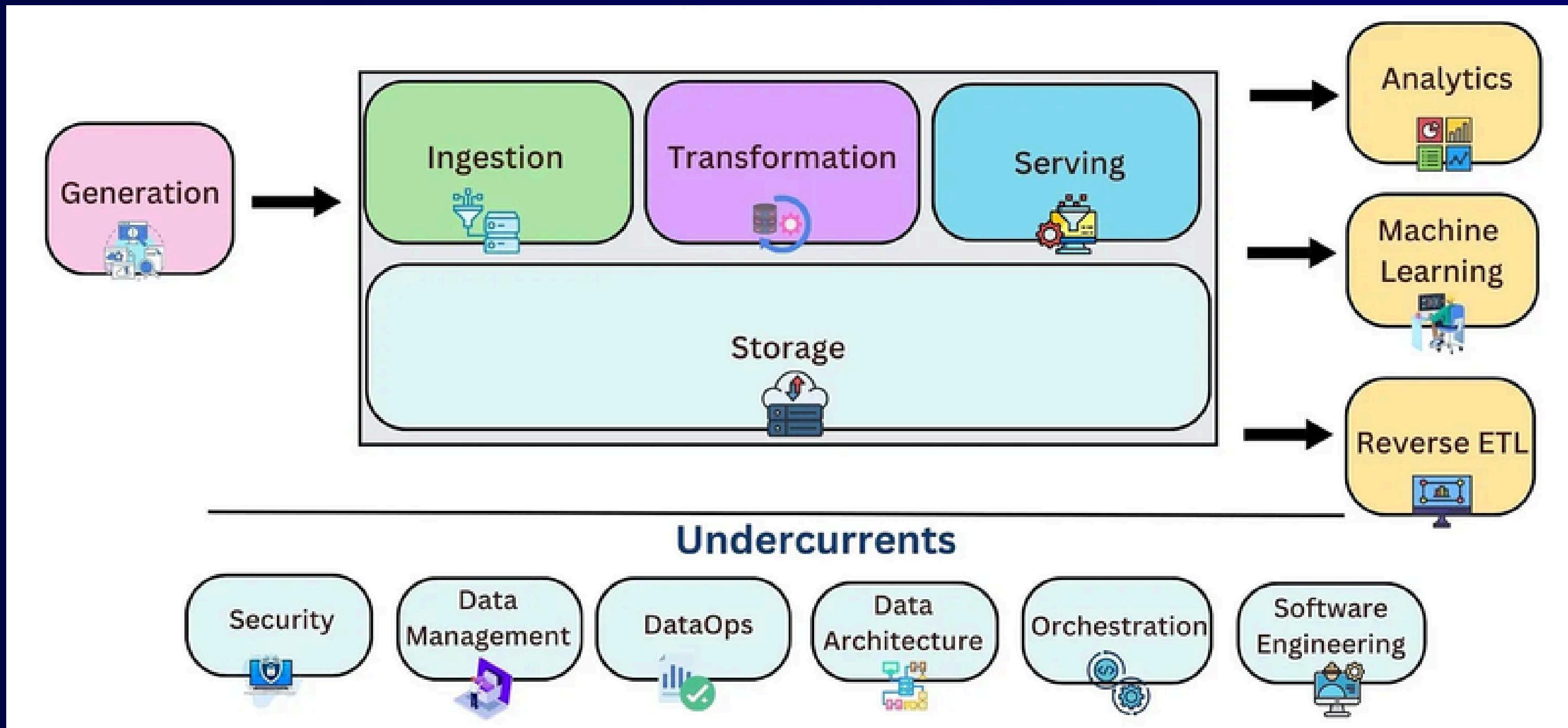
Là sự kết hợp của kiến trúc, hệ thống và quy trình di chuyển dữ liệu qua các giai đoạn trong vòng đời của data engineering.

# Vòng đời của một Data Engineer



Công việc của một Data Engineering là lấy **dữ liệu thô** từ đâu đó, biến chúng thành những thứ **hữu dụng**, và làm cho chúng **luôn sẵn sàng** cho các công việc đằng sau!

# Vòng đời của một Data Engineer



\*Undercurrents: Yếu tố ngầm, ảnh hưởng ngầm

# Những bên liên quan



Analysts



Data Scientist



Machine Learning  
Engineers



Salespeople



Marketing  
Professionals



Executives

# Stakeholders

## Upstreams



Software Engineers

- Khối lượng dữ liệu?
- Tần suất?
- Định dạng dữ liệu?
- Bảo mật dữ liệu
- Tuân thủ quy định



Data Engineer

## Downstream



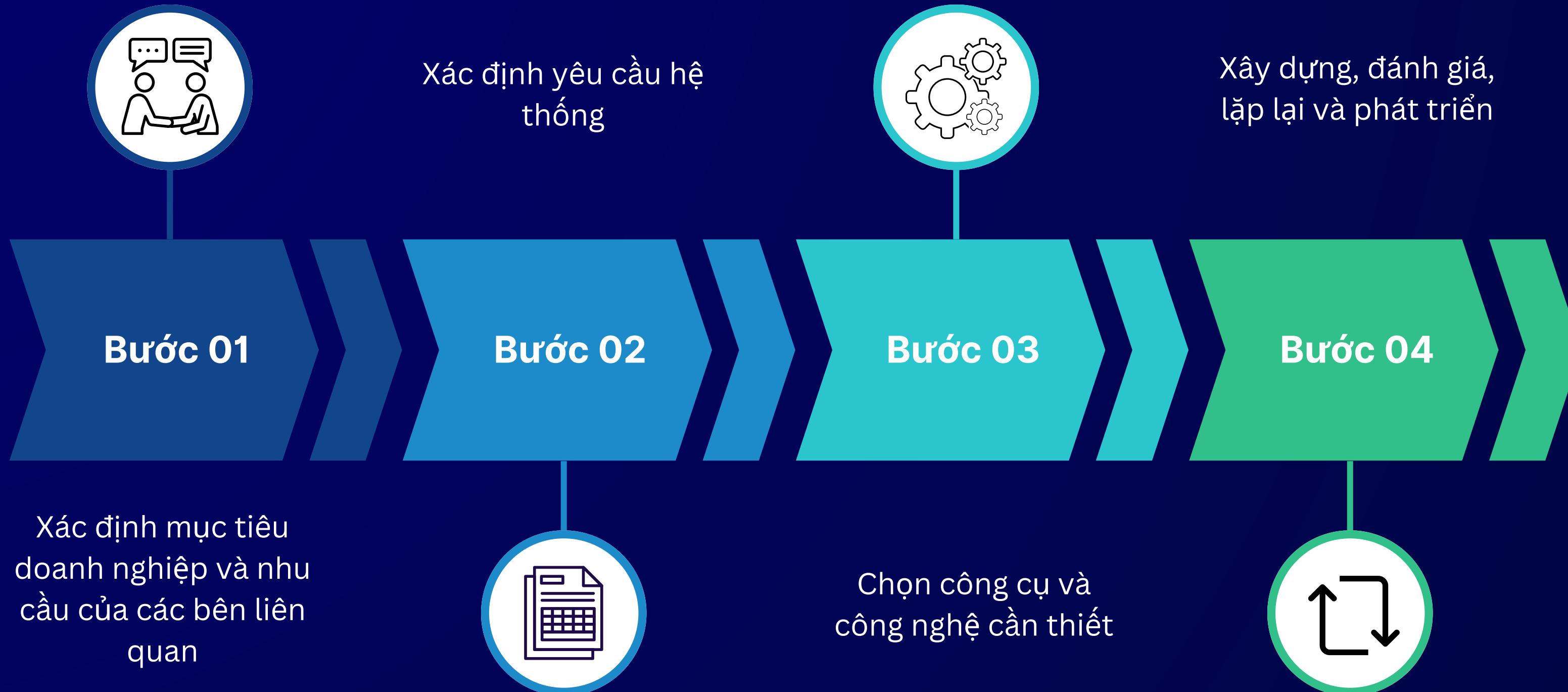
Analysts

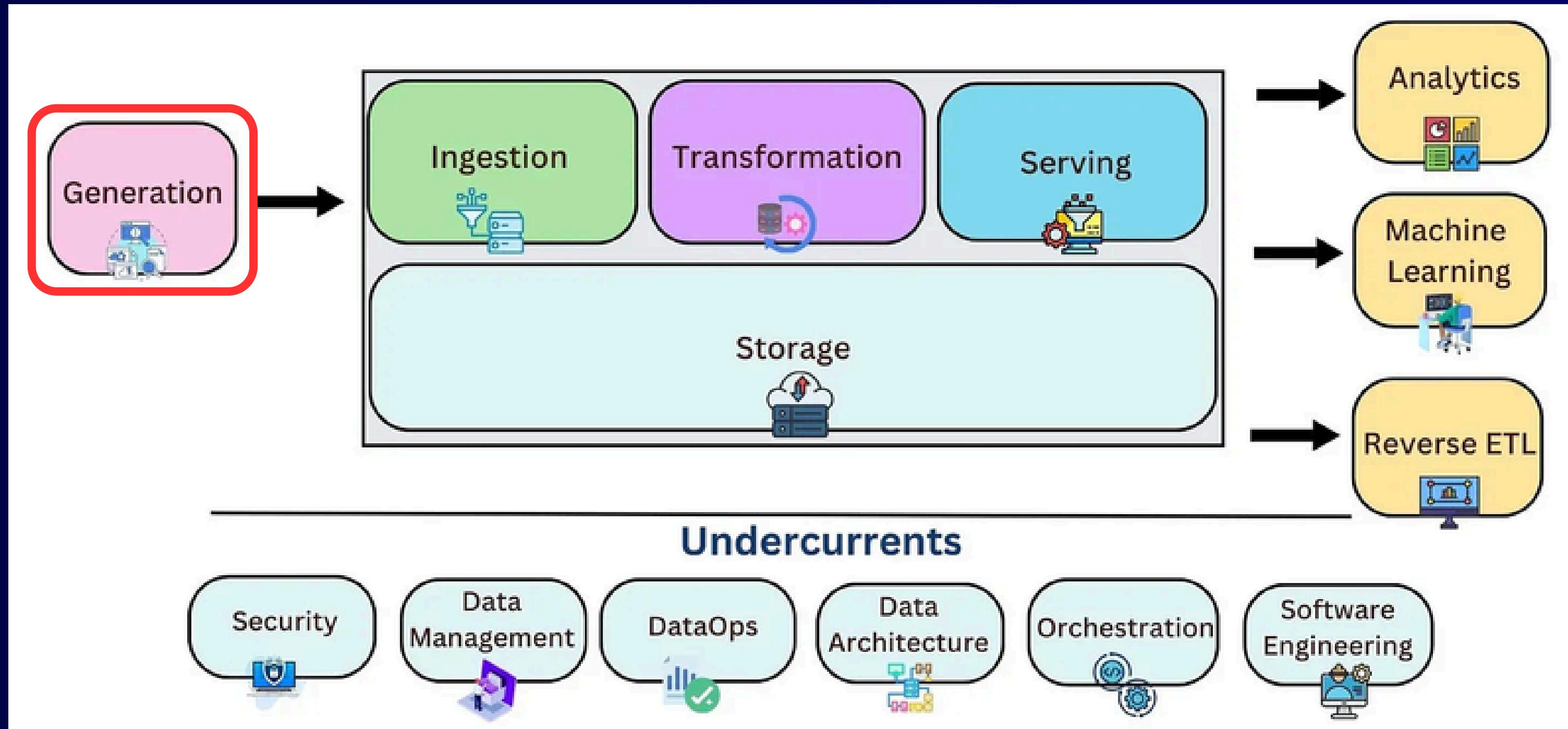
- Bao lâu một lần?
- Những thông tin gì?
- Độ trễ là bao nhiêu?

Query



# Suy nghĩ như một Data Engineer



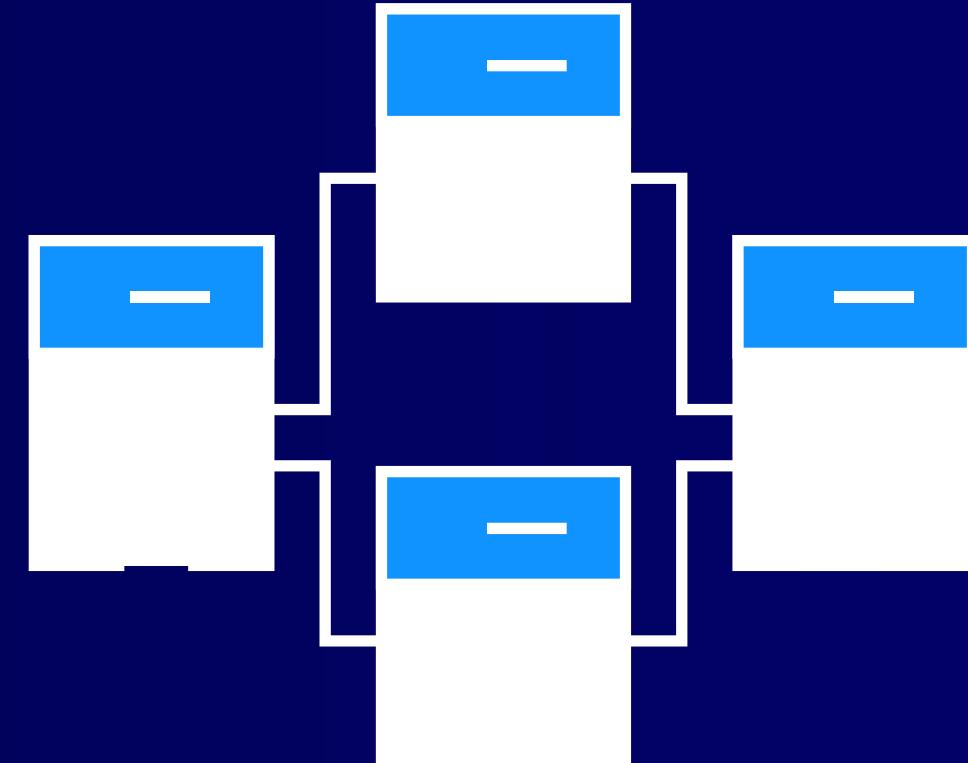


# Source Systems - Databases

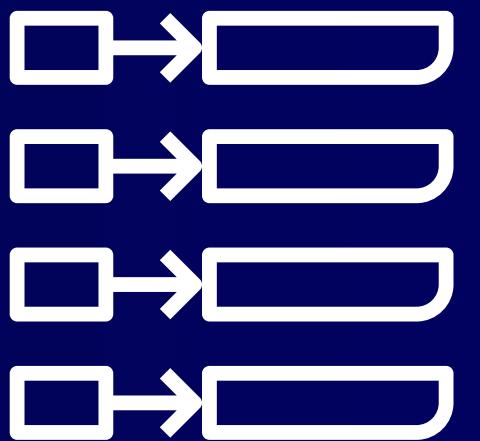


**Databases**

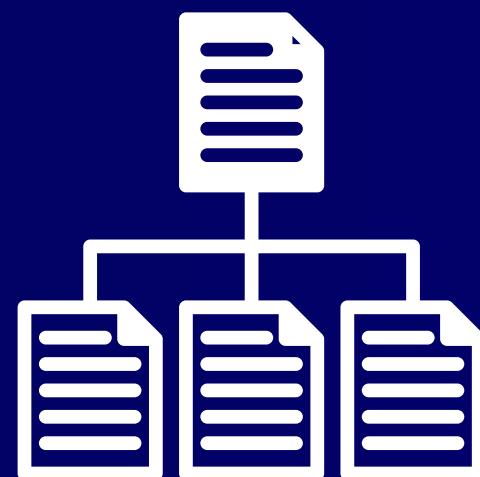
Relational Databases



NoSQL Databases



Key-Value

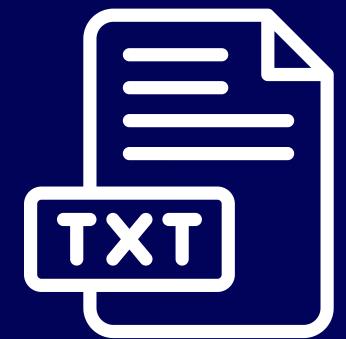


Document Stores

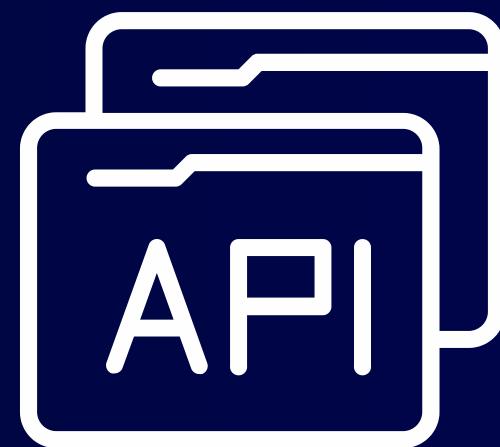
# Source Systems - Files



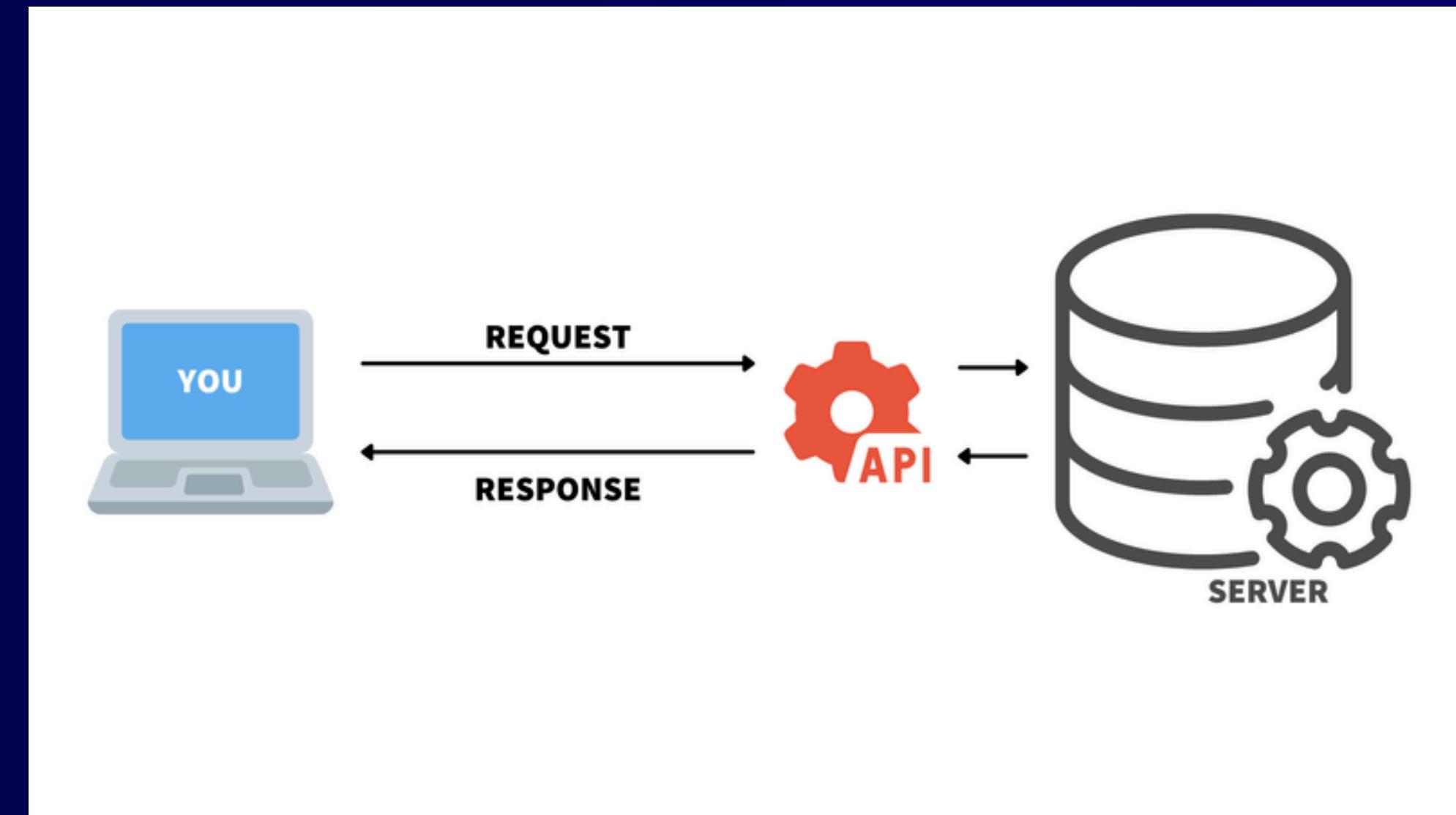
Files

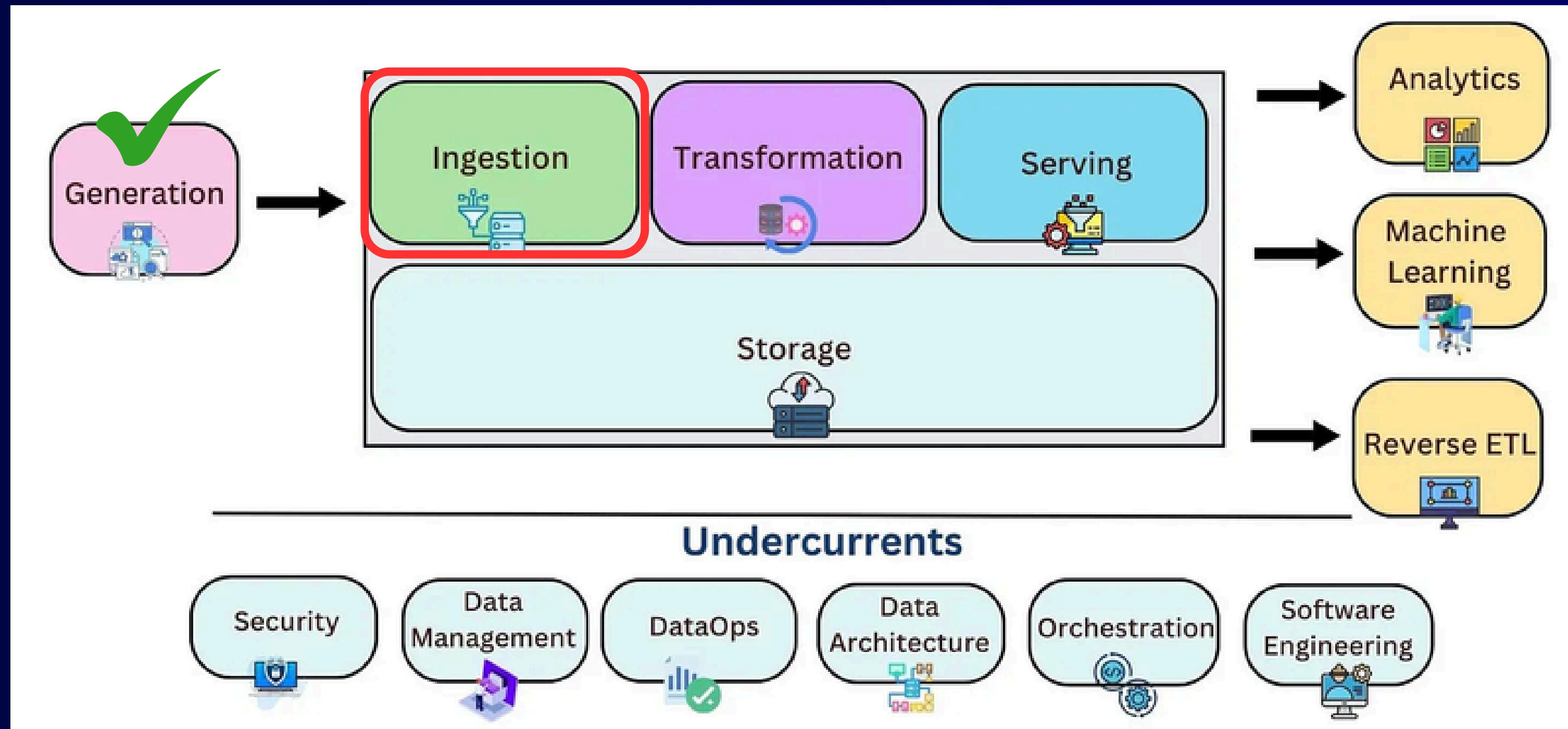


# Source Systems - API

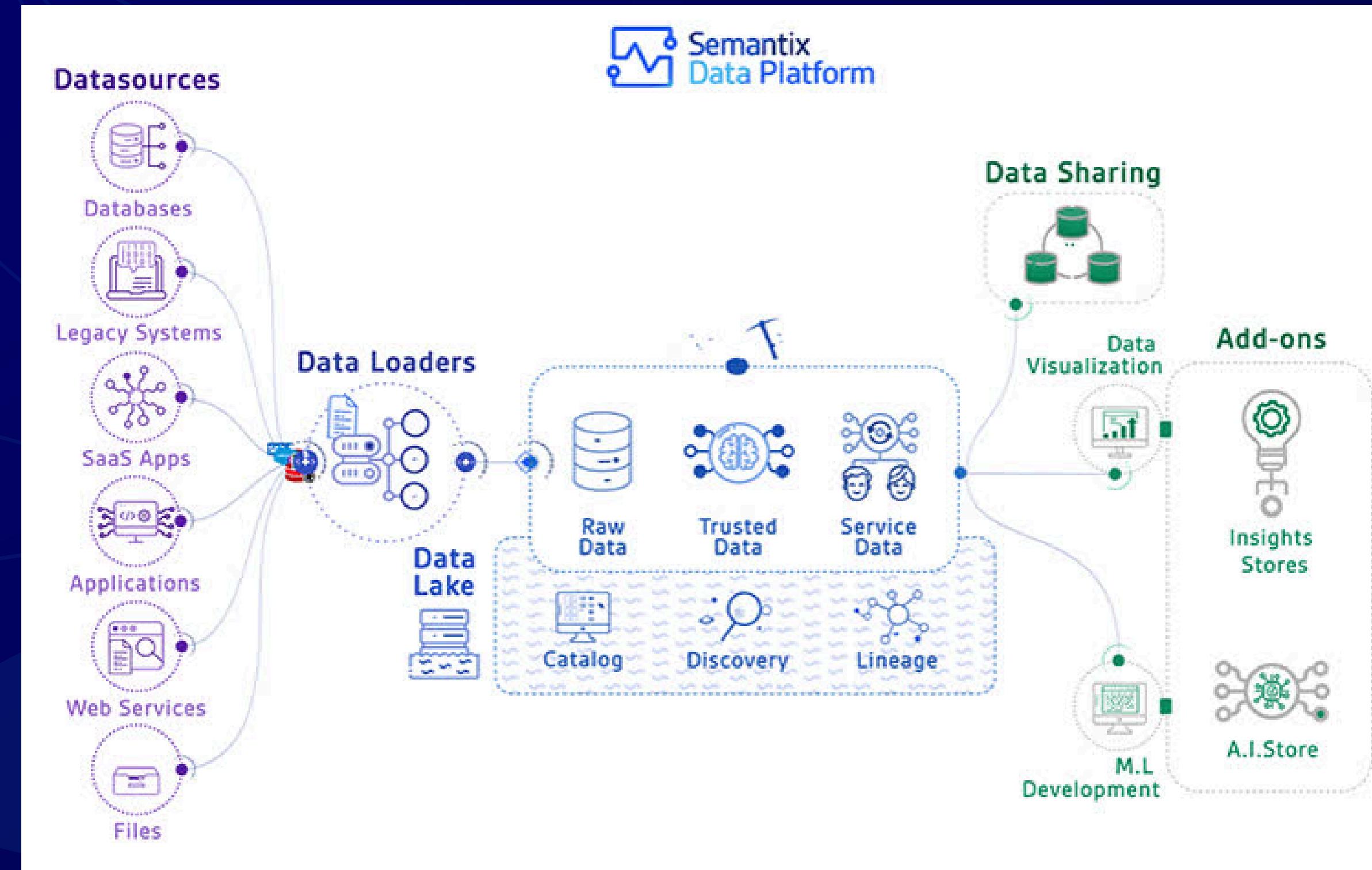


API





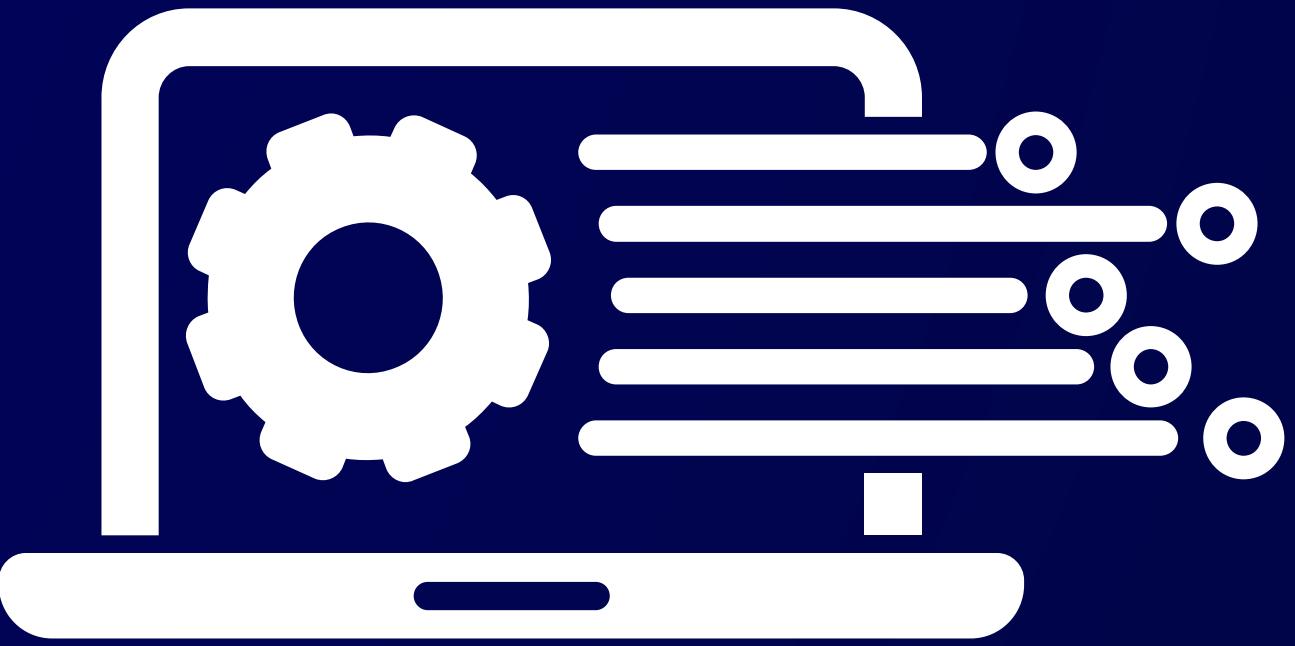
# Ingestion - Thu thập dữ liệu



# Tần suất thu thập dữ liệu



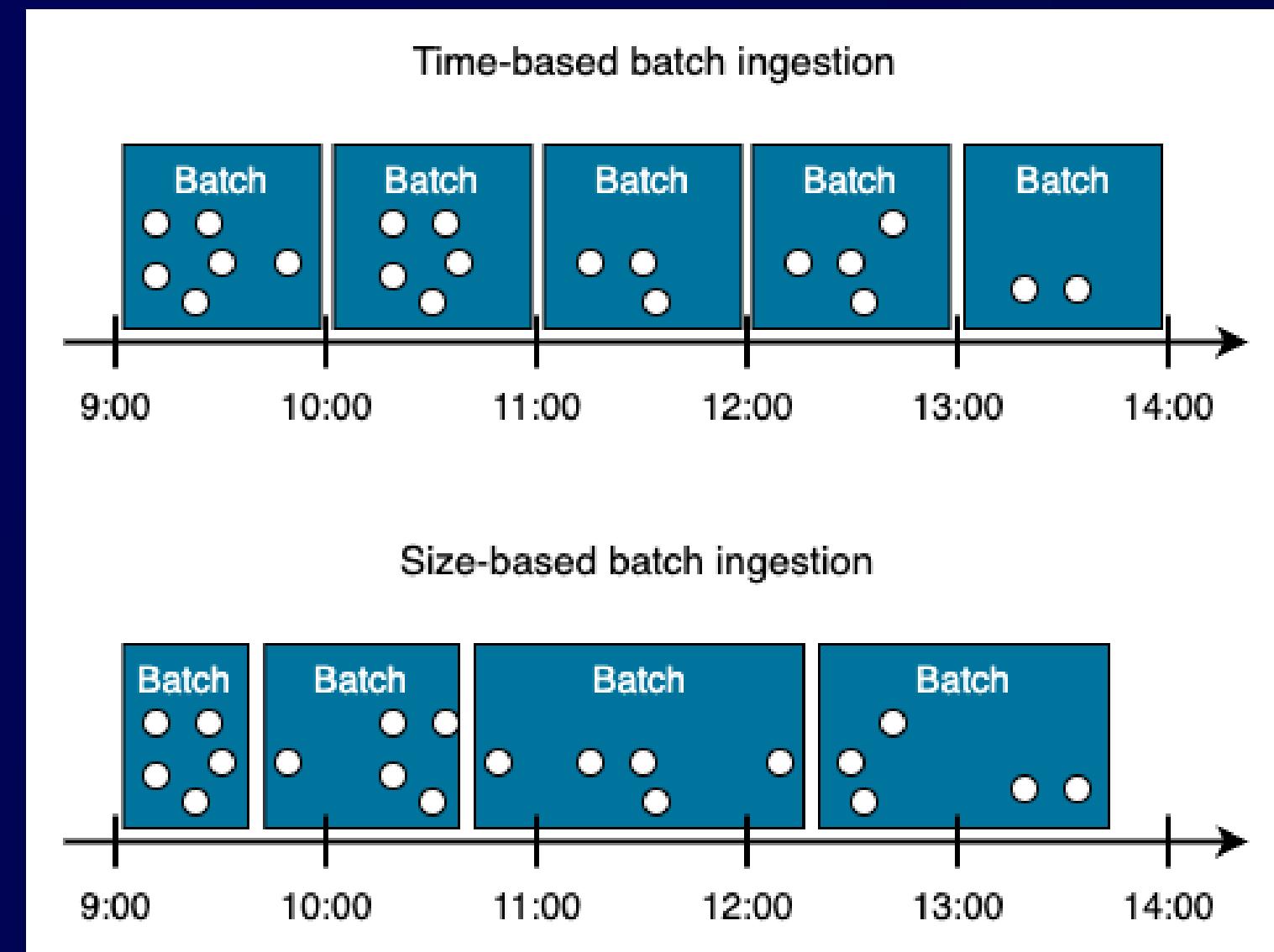
**Batch Ingestion**



**Streaming Ingestion**

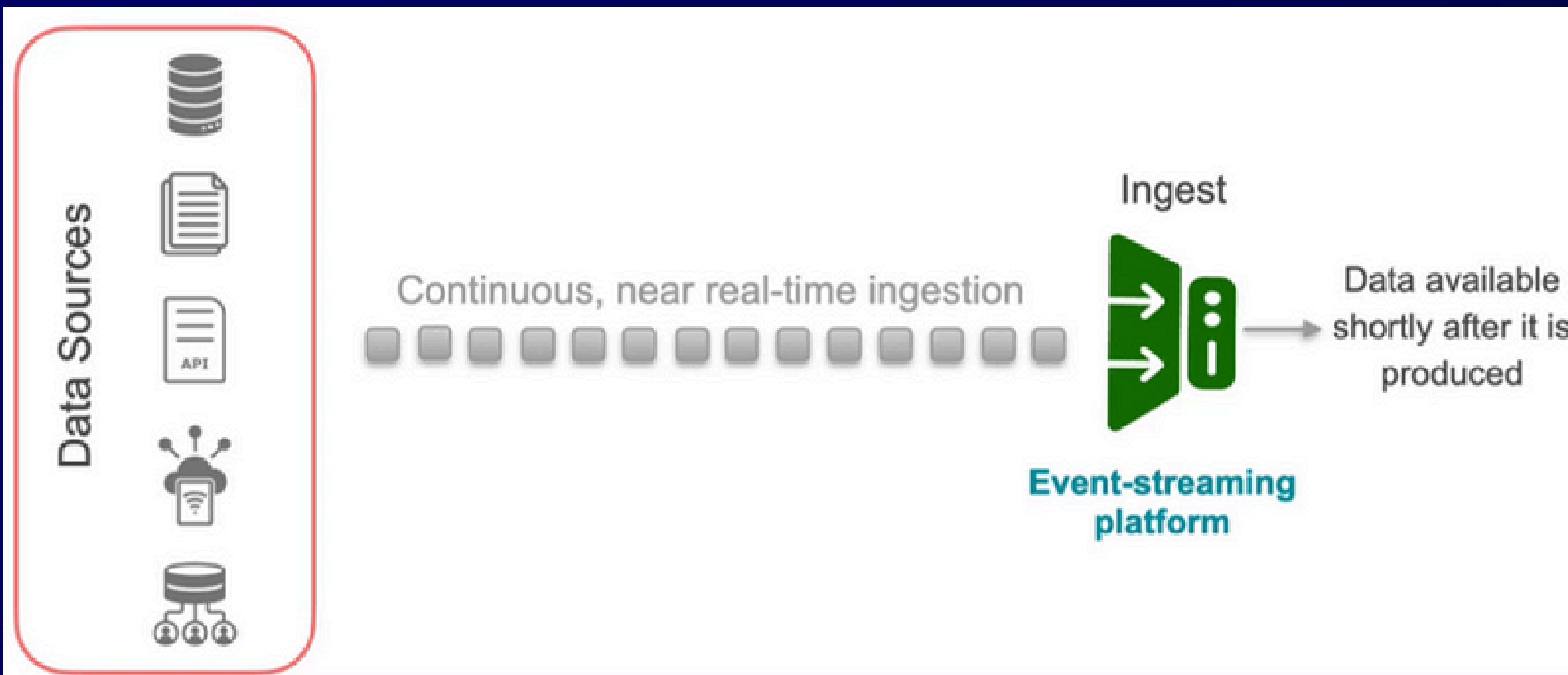
# Batch Ingestion

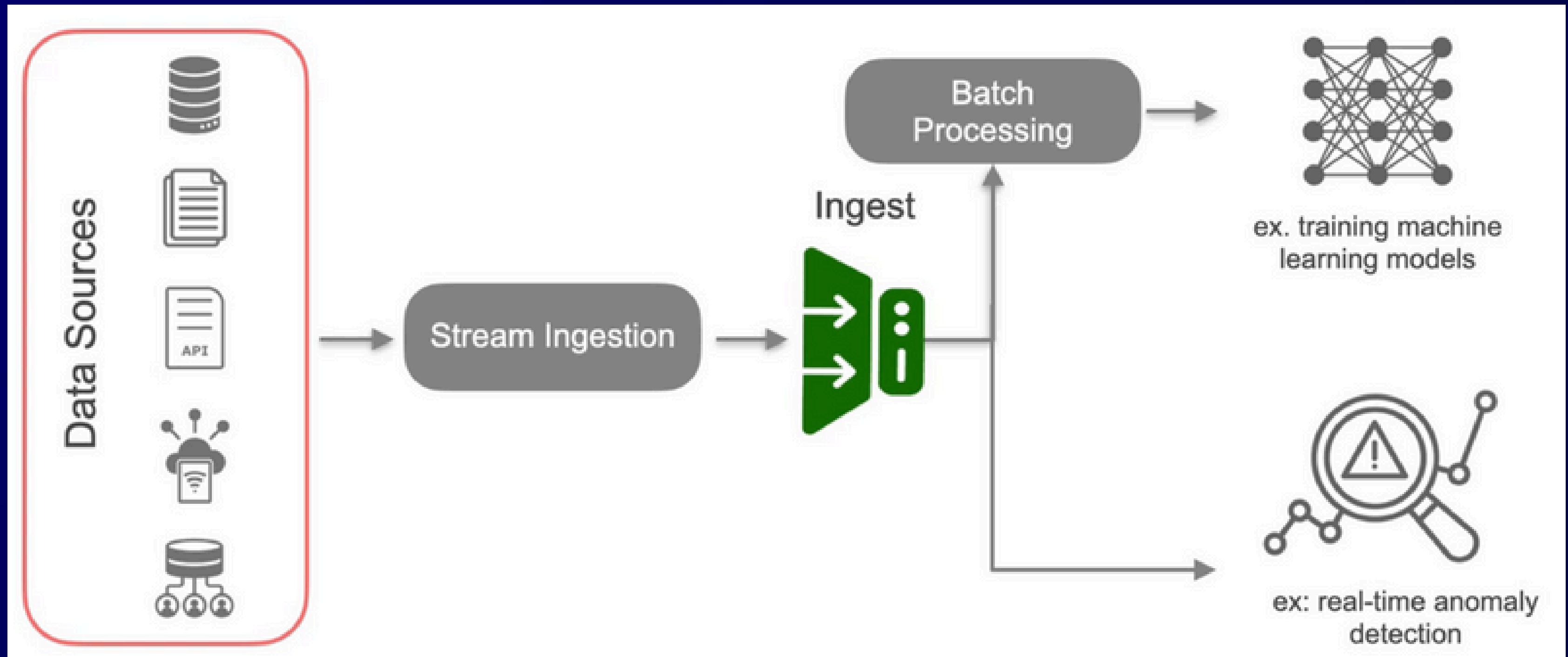
- Là quá trình thu thập và đưa dữ liệu vào một hệ thống lưu trữ hoặc xử lý theo **từng khối lượng lớn, rời rạc, được gọi là "lô" (batch), tại các thời điểm hoặc khoảng thời gian nhất định.**
- Ví dụ:
  - Tải lên các tệp log từ server web vào cuối mỗi ngày để phân tích.
  - Nhập dữ liệu giao dịch của ngày hôm trước từ hệ thống POS vào kho dữ liệu vào mỗi buổi sáng.
  - Sao chép toàn bộ cơ sở dữ liệu từ hệ thống nguồn sang hệ thống đích hàng tuần.

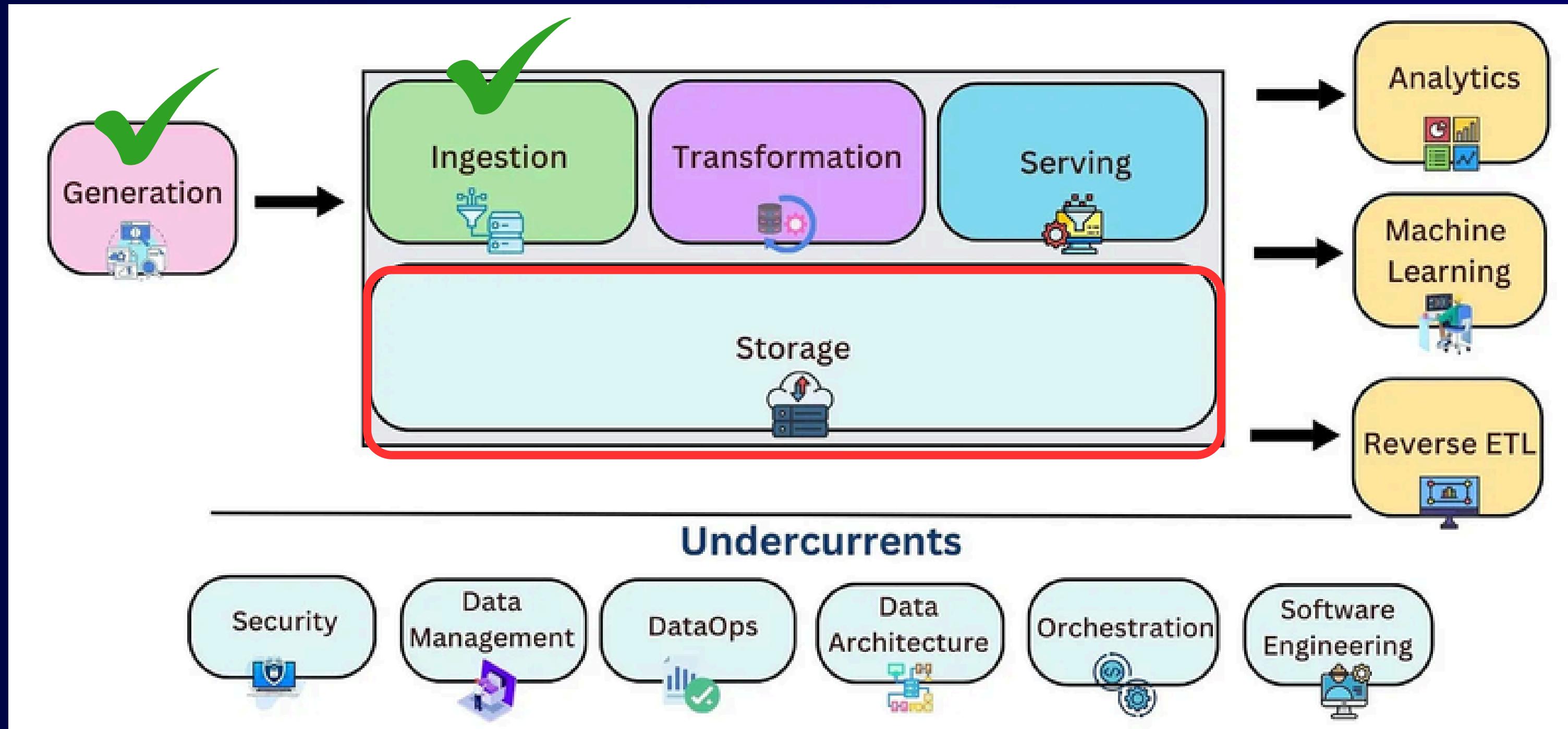


# Streaming Ingestion

- Là quá trình thu thập và đưa dữ liệu vào một hệ thống lưu trữ hoặc xử lý liên tục, **gần như theo thời gian thực, ngay khi dữ liệu đó được tạo ra hoặc phát sinh.**







# Những thành phần phần cứng thô

Solid-state storage



Magnetic disk



Magnetic disk (Đĩa từ)

- Là xương sống của hệ thống lưu trữ dữ liệu hiện đại.
- Rẻ hơn gấp 2-3 lần so với solid-state storage (lưu trữ thể rắn - sử dụng chip nhớ để đọc, ghi dữ liệu)

# Những thành phần phần cứng thứ

## RAM (Random Access Memory)

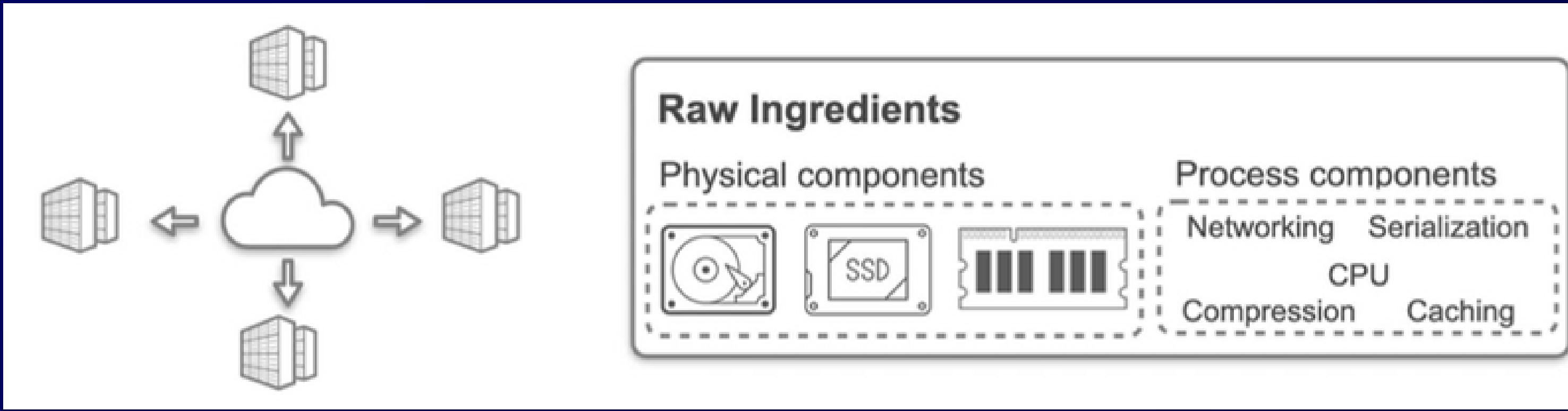
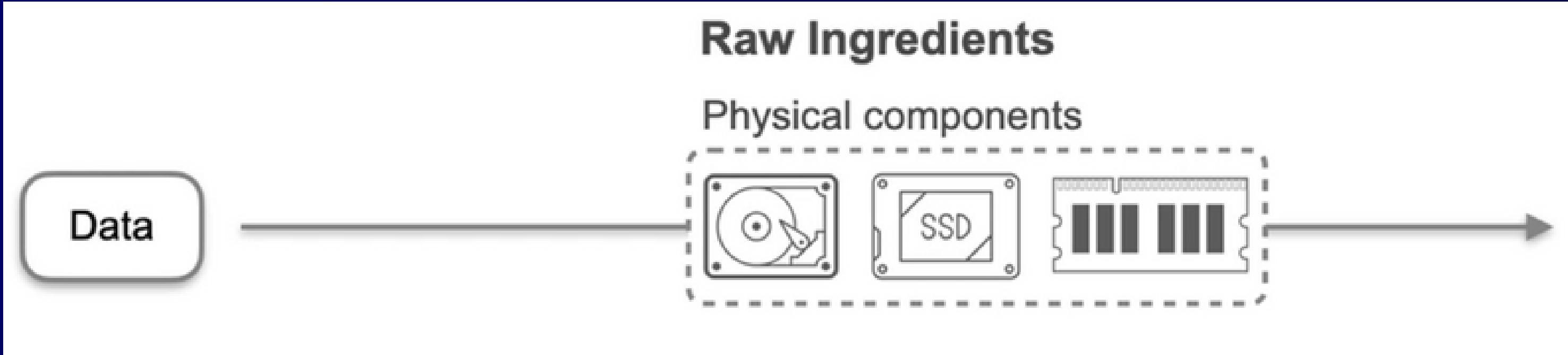


RAM

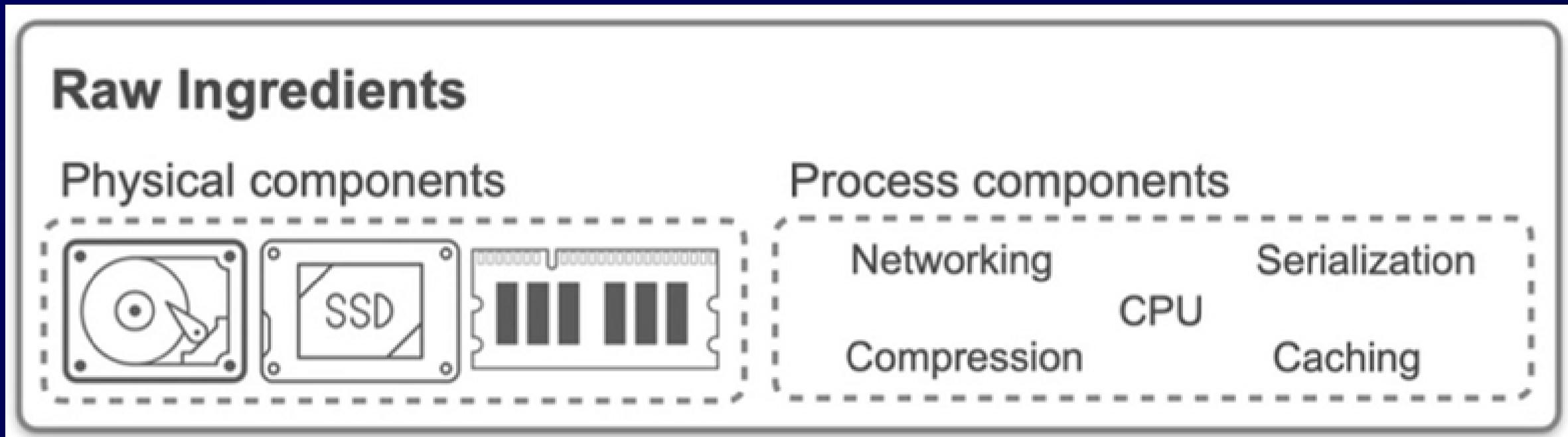
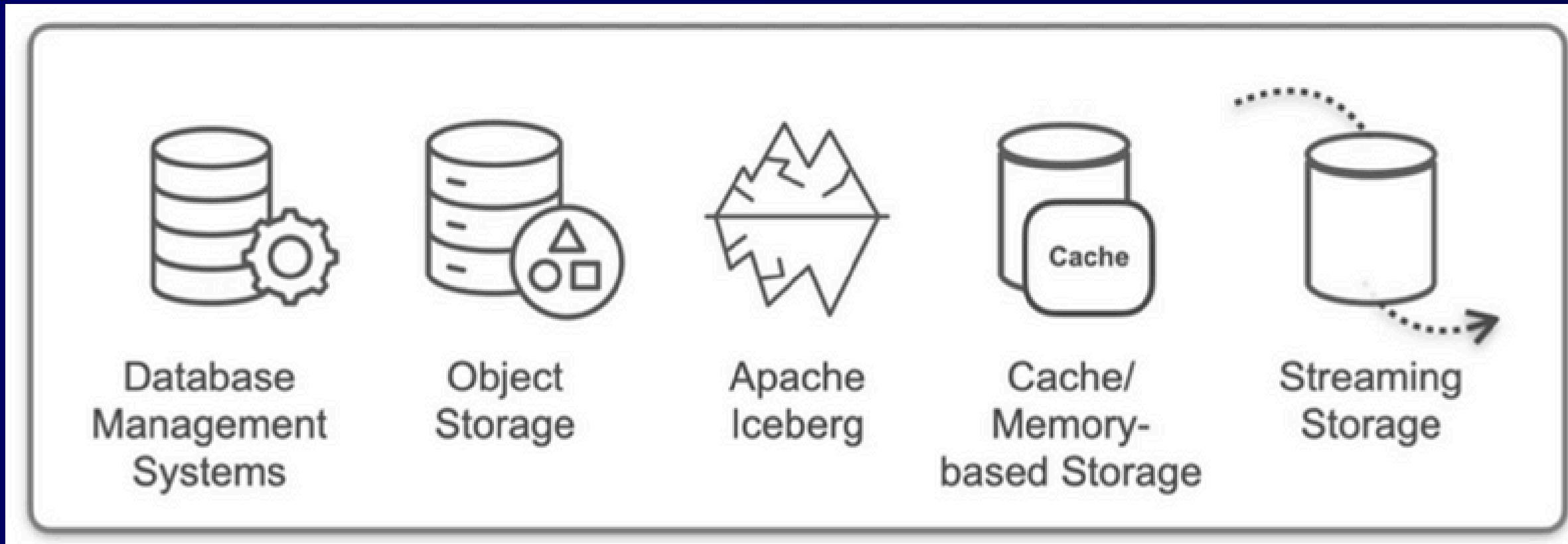
- Tốc độ đọc / ghi nhanh hơn
- Đắt hơn 30 - 50 lần so với solid-state storage
- Dữ liệu sẽ mất khi không có điện.



# Storage

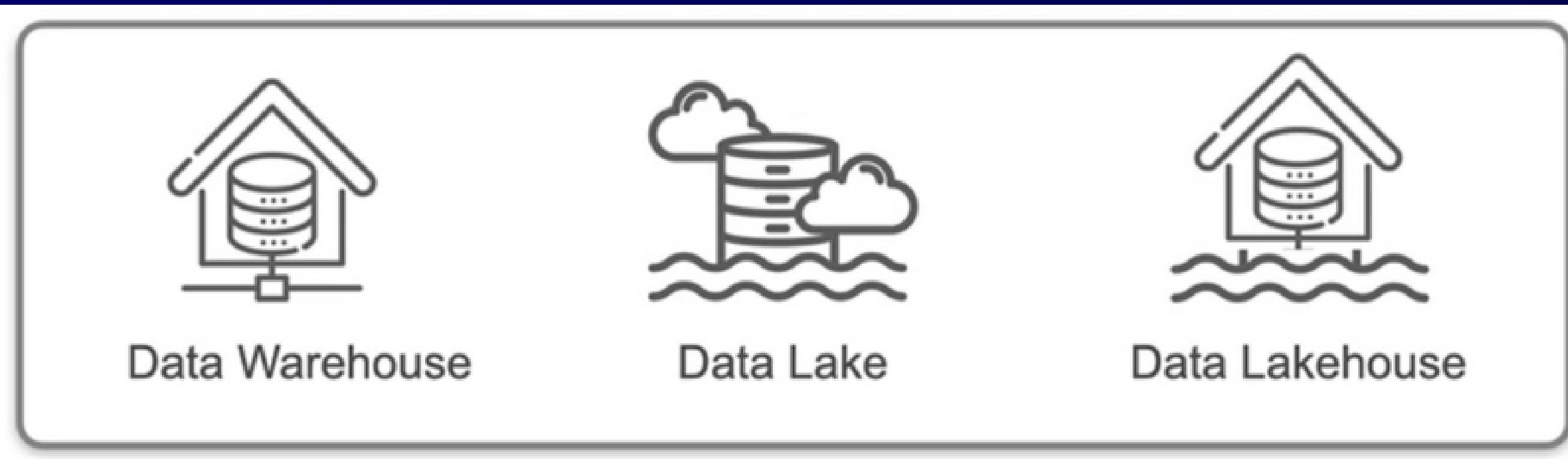


# Storage Systems



# Storage Abstractions

(Sự kết hợp của các hệ thống lưu trữ)

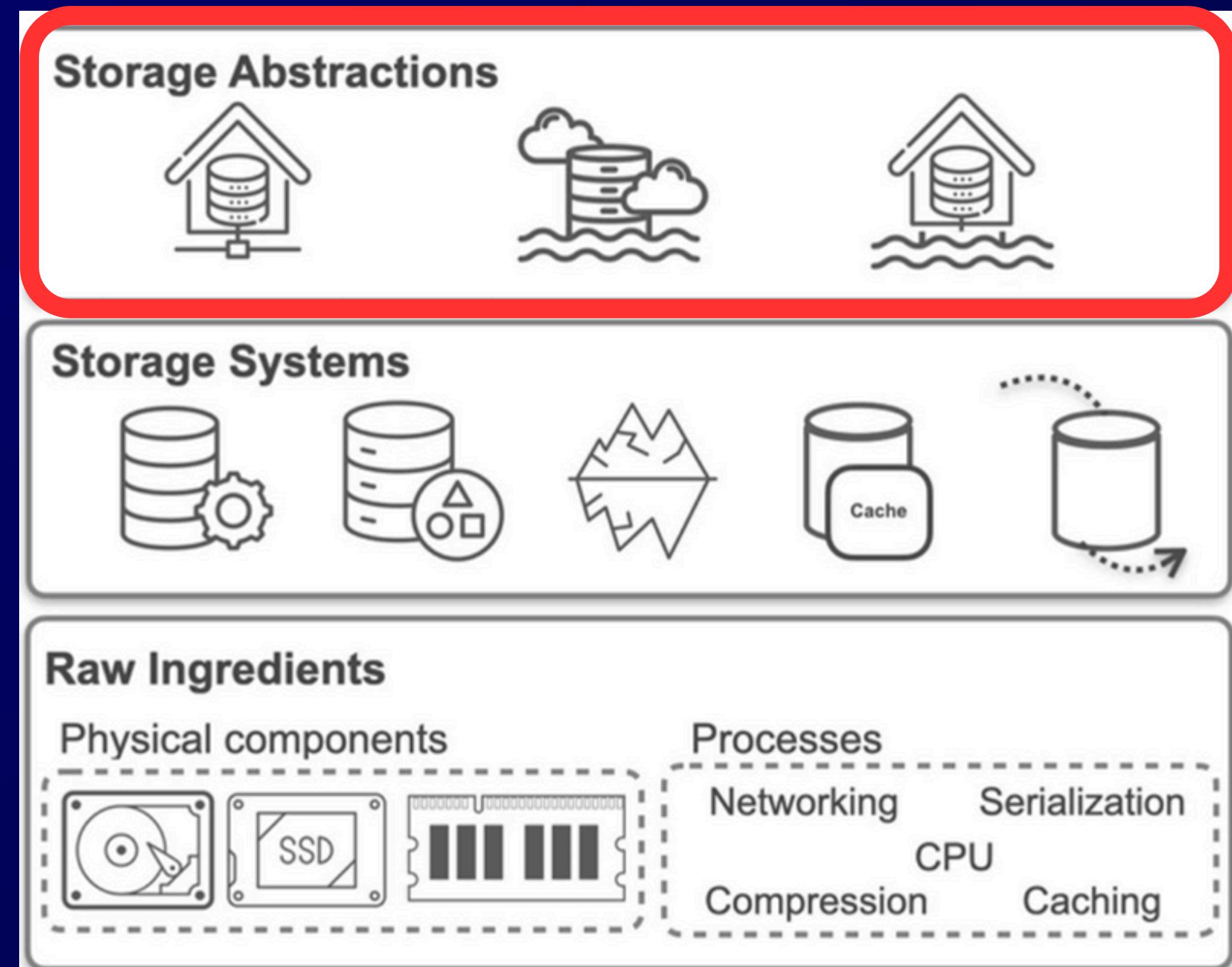


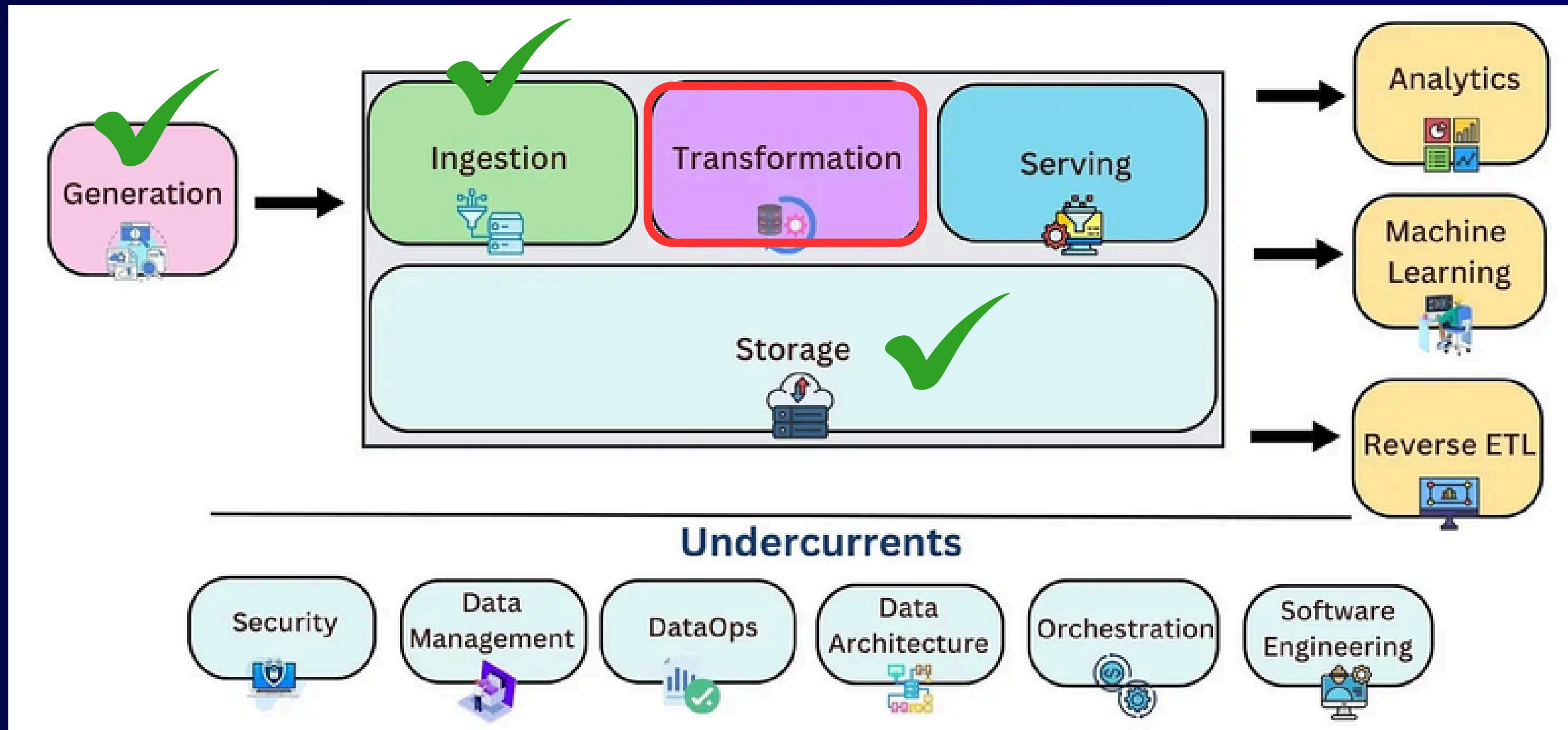
Việc lựa chọn giữa các hệ thống lưu trữ này  
phụ thuộc vào nhu cầu về:

- Độ trễ - Latency
- Tính mở rộng - Scalability
- Chi phí - Cost

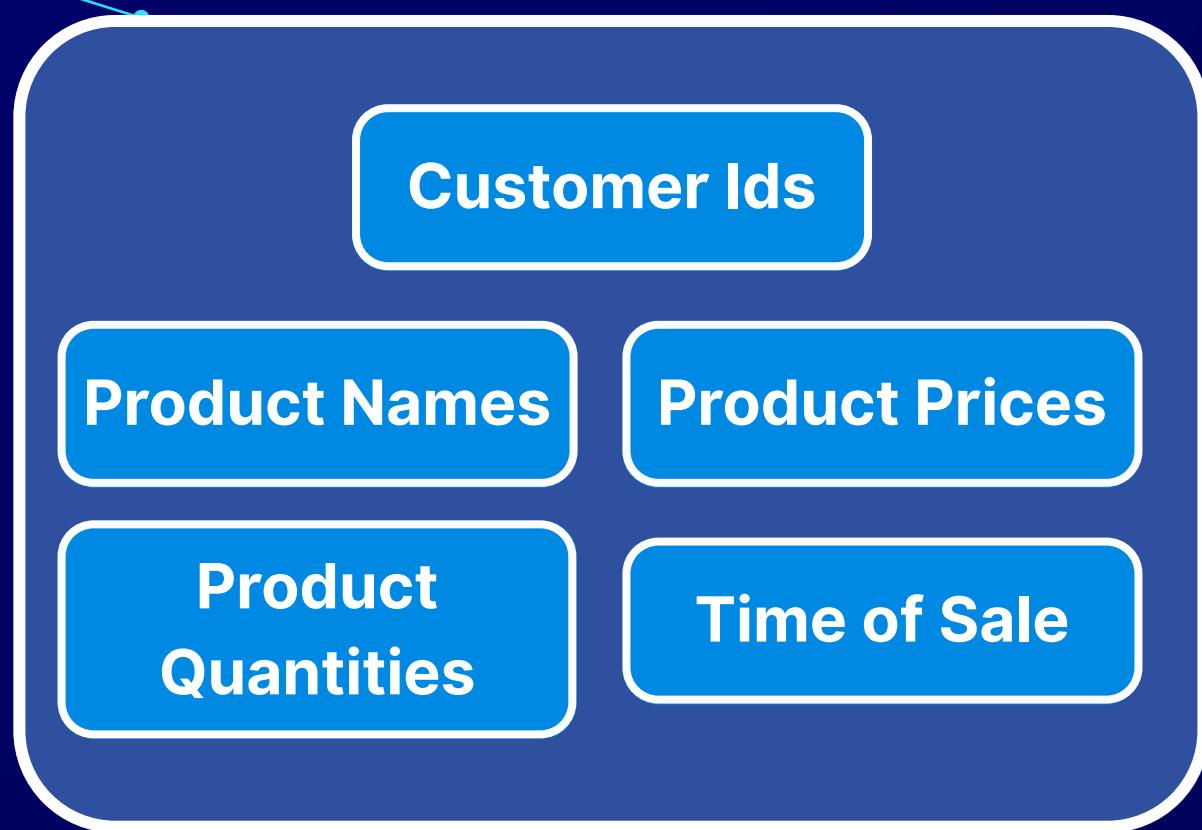
# Storage Hierarchy

(Phân cấp lưu trữ)

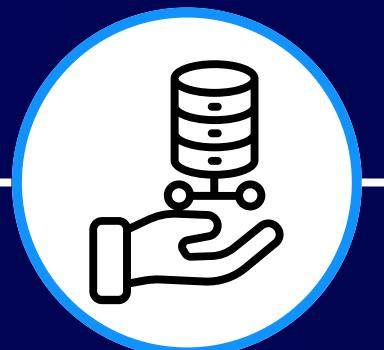
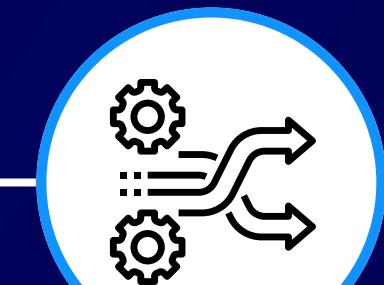




# Transformation



Transform



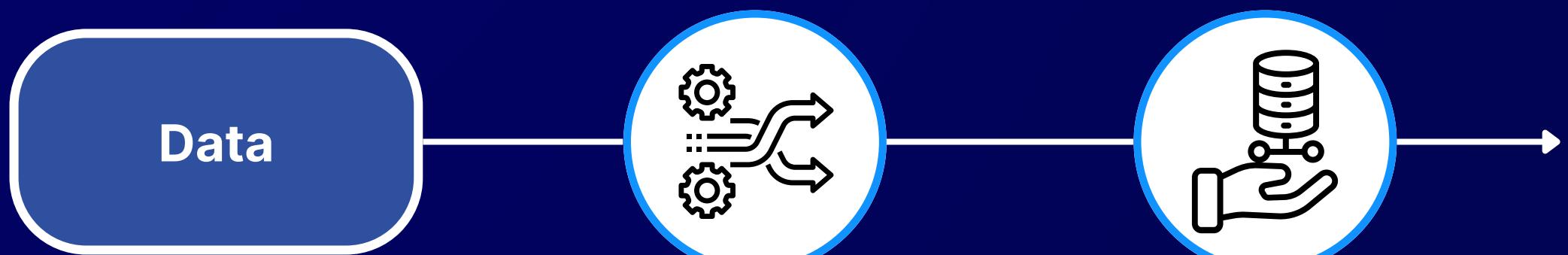
Serve



**Business Analyst**

Mục tiêu: báo cáo doanh số  
hằng ngày của một số sản  
phẩm

# Transformation



## Transform

Chuyển đổi thành  
các cấu trúc và  
tạo đặc trưng.



## Data Scientist

Mục tiêu: Phân tích dự  
đoán

## Query

Đưa ra yêu cầu để đọc các bản ghi trong một database hoặc hệ thống lưu trữ khác.



Data Warehouse

Query

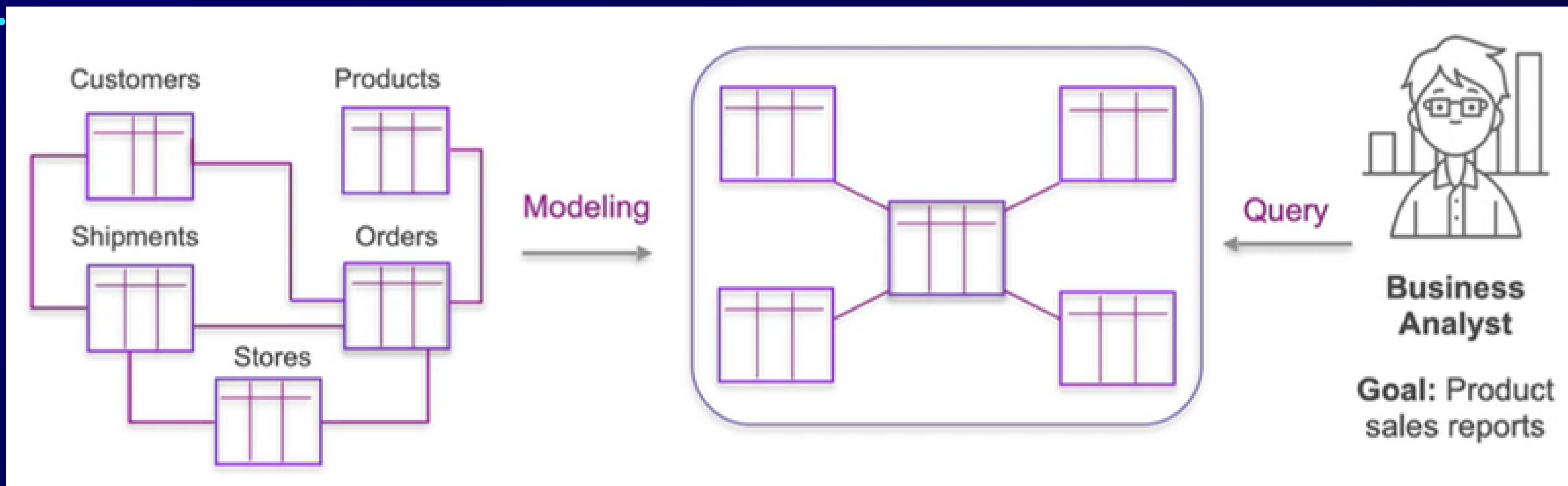
- Dữ liệu dạng bảng
- Dữ liệu bán cấu trúc



Query Language

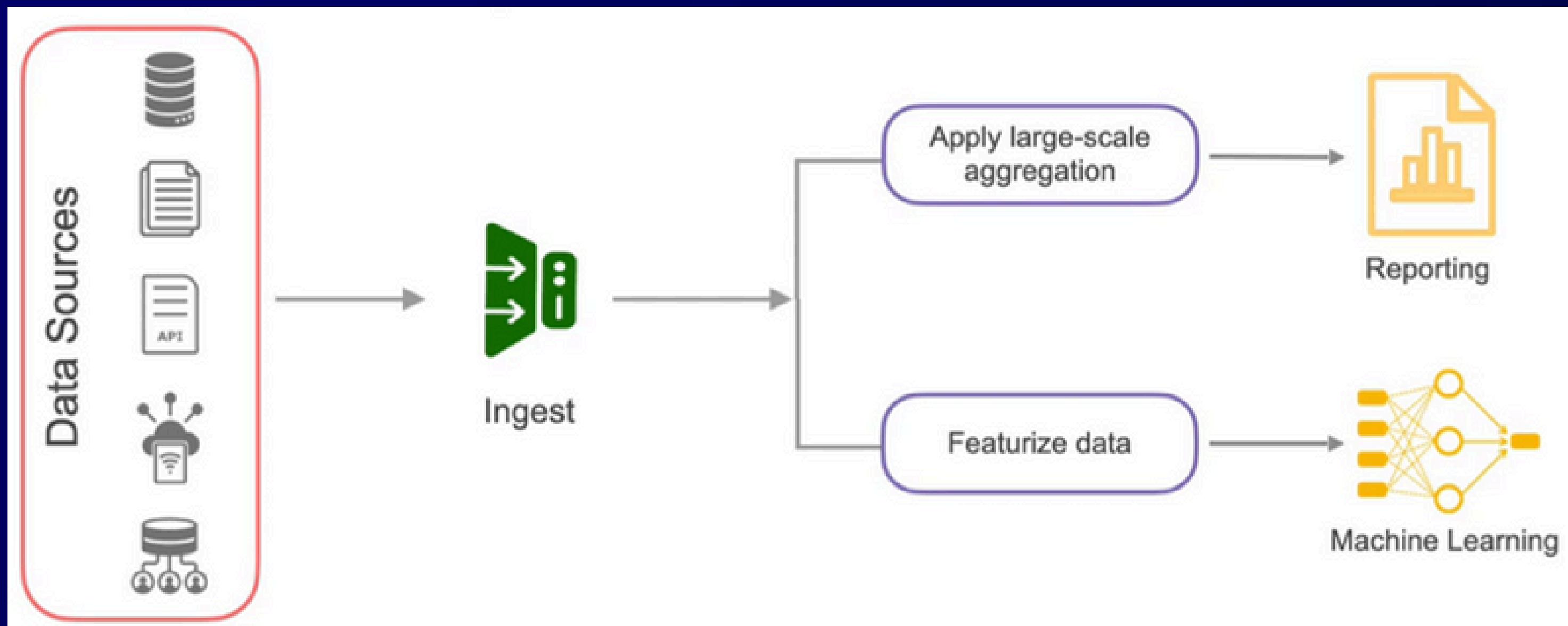
## Data modeling

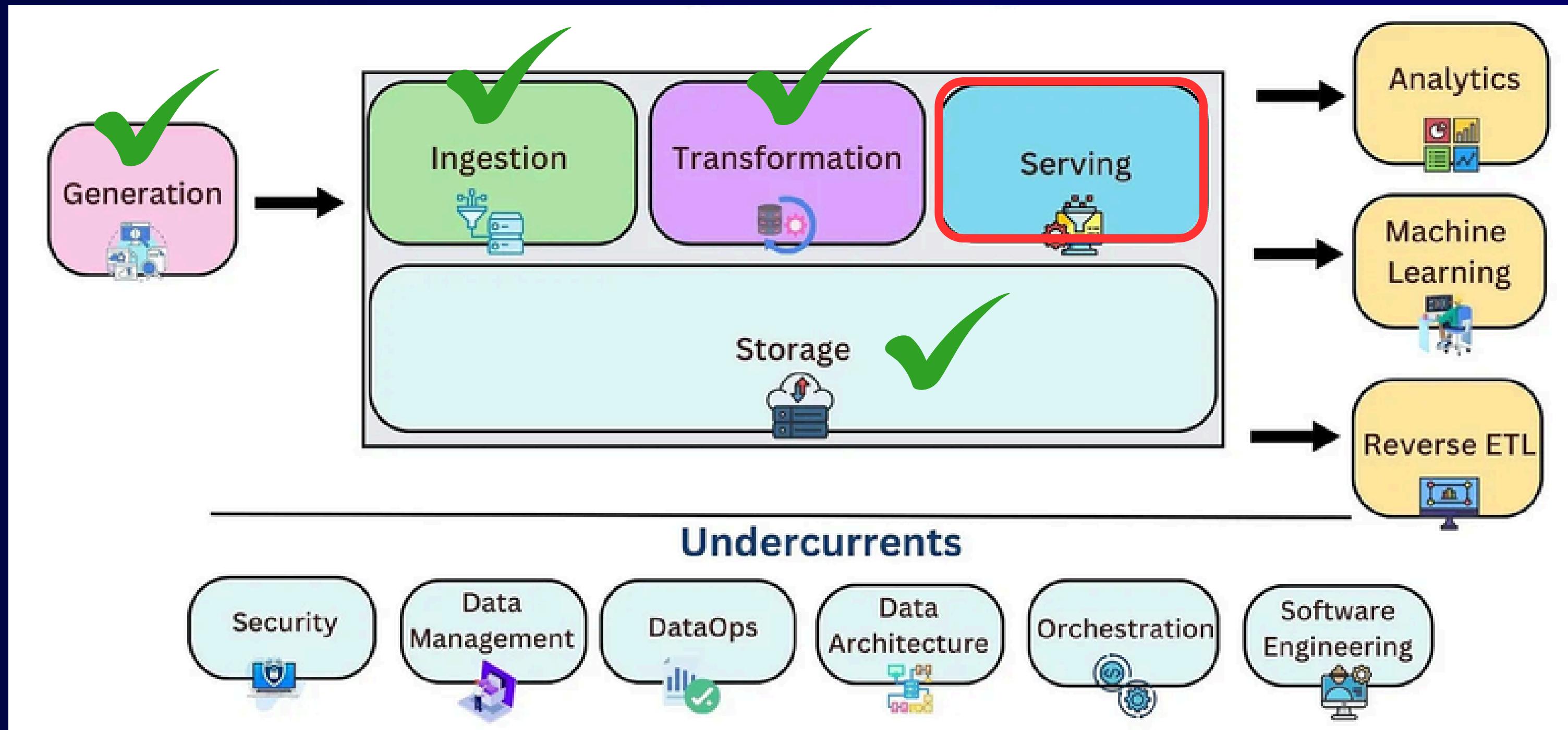
Chọn một cấu trúc mạch lạc, rõ ràng cho dữ liệu  
để làm cho chúng hữu dụng với doanh nghiệp



## Data transformation

Là những thao tác với dữ liệu, cải tiến và lưu lại để sử dụng cho những công việc phía sau





# **Analytics**

Analytics (Phân tích) là quá trình khám phá những hiểu biết cốt lõi và các quy luật ẩn chứa bên trong dữ liệu.

**Business Intelligence**

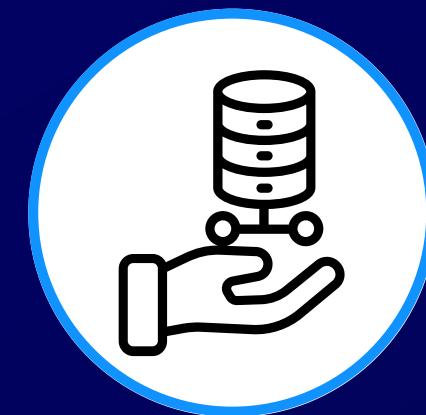
**Operational Analytics**

**Embedded Analytics**

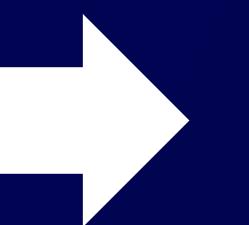
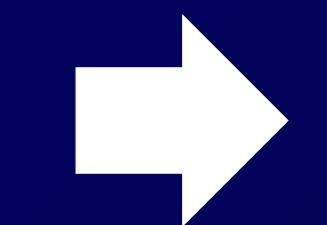
# Analytics

Business Intelligence

Khám phá dữ liệu từ lịch sử tới hiện tại để khám phá những thông tin có giá trị.



Serve



Reports



Dashboards



# Analytics

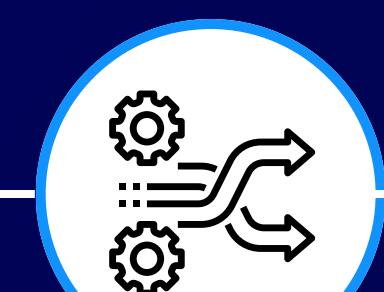
## Operational Analytics

Giám sát dữ liệu thời gian thực để có thể thực hiện những hành động ngay lập tức

### E-commerce website



Ingest



Transform



Serve



Theo dõi số liệu hiệu suất website thời gian thực

# Analytics

## Embedded Analytics

Phân tích bên ngoài hoặc phục vụ khách hàng

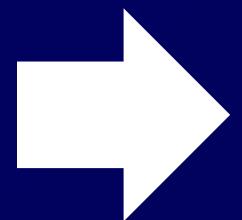


Bảng tài chính cá nhân phục  
vụ khách hàng

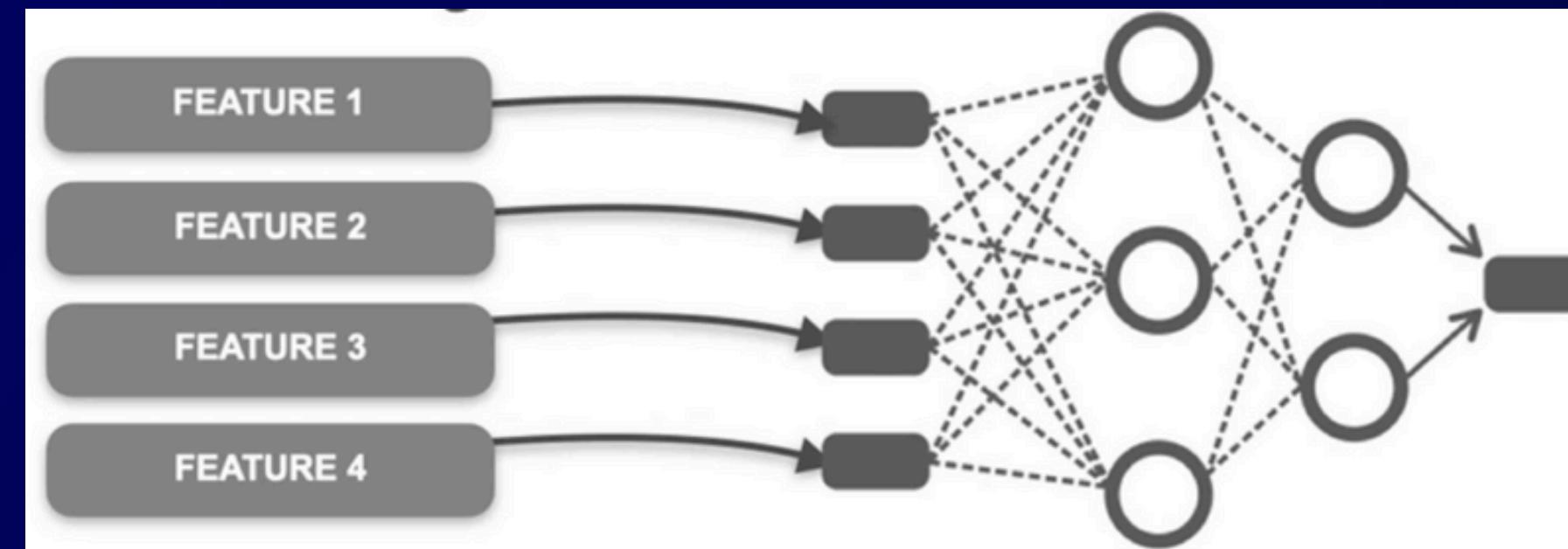
# Machine Learning



Serve

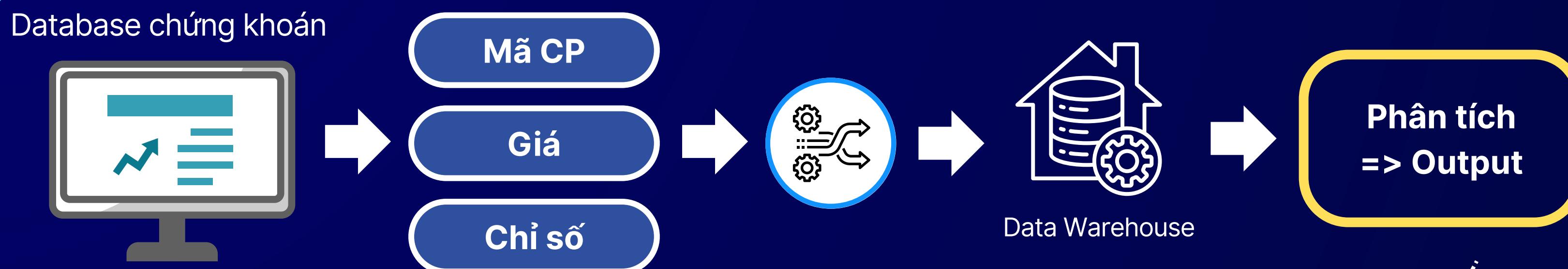


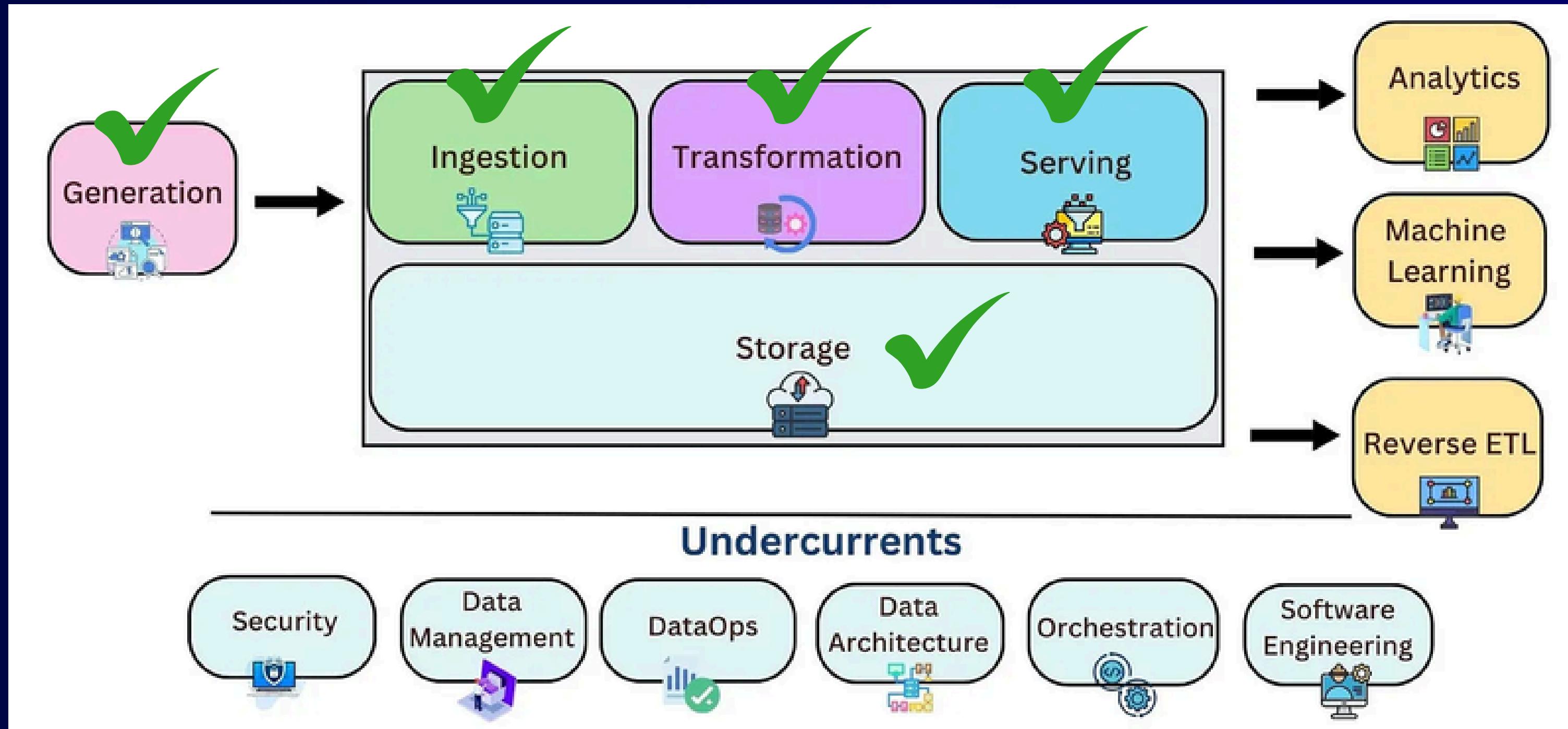
- Huấn luyện mô hình



- Suy luận, đưa ra kết quả thời gian thực
- Theo dõi dữ liệu lịch sử và nguồn gốc

# Reverse ETL





# SOURCE SYSTEM

---

# Các loại hệ thống nguồn

---

## Structured Data

Là dữ liệu được tổ chức theo một bảng với các cột và hàng

The diagram illustrates a structured data table with 5 rows and 4 columns. The columns are labeled 'ID', 'Last', 'First', and 'Card'. The first column ('ID') contains values 14, 25, 14, and 25. The second column ('Last') contains values Barry, Goode, Barry, and Goode. The third column ('First') contains values John, Cynthia, John, and Cynthia. The fourth column ('Card') contains values XXX878, XXX980, XXX978, and XXX990. A yellow double-line border highlights the first two rows (ID 14 and 25). An orange double-line border highlights the last two rows (ID 14 and 25). A grey arrow points from the right side of the table towards the word 'Rows'. A grey arrow points downwards from the bottom of the table towards the word 'Columns'.

ID	Last	First	Card
14	Barry	John	XXX878
25	Goode	Cynthia	XXX980
14	Barry	John	XXX978
25	Goode	Cynthia	XXX990

Rows

Columns

## Structured Data

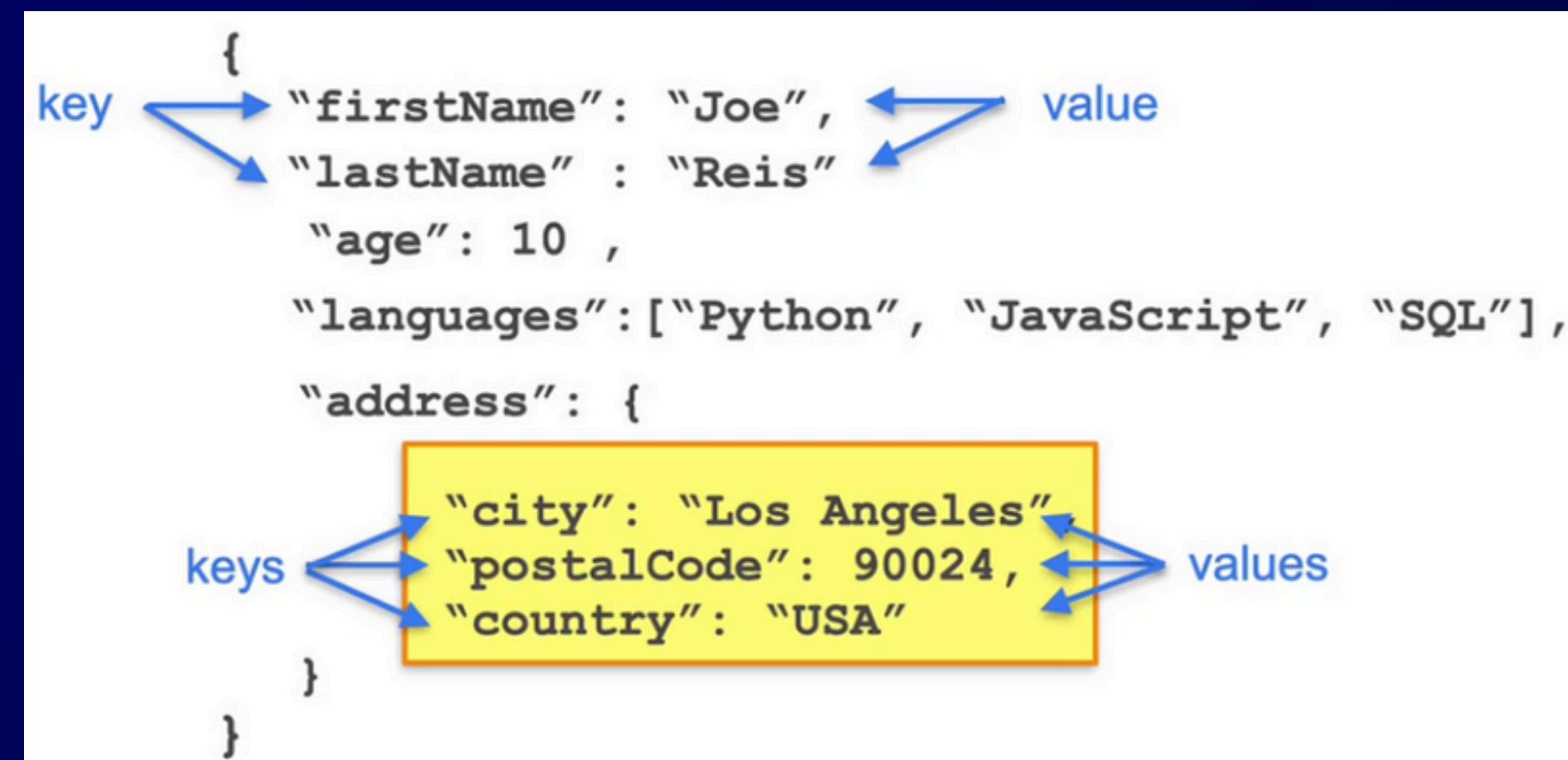
Là dữ liệu được tổ chức theo một bảng với các cột và hàng

## Semi-Structured Data

Là dữ liệu không tổ chức theo bảng nhưng vẫn có cấu trúc

# JavaScript Object Notation (JSON)

Một loạt các cặp key-value



## **Structured Data**

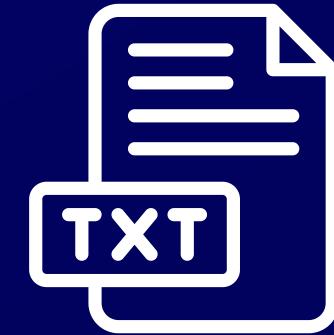
Là dữ liệu được tổ chức theo một bảng với các cột và hàng

## **Semi-Structured Data**

Là dữ liệu không tổ chức theo bảng nhưng vẫn có cấu trúc

## **Unstructured Data**

Là dữ liệu không tổ chức theo một cấu trúc nào



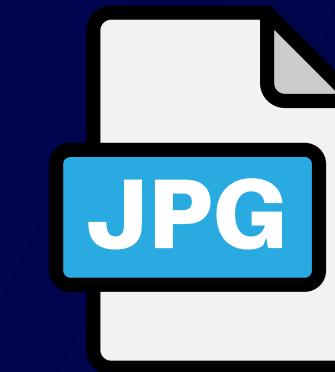
Text



Audio



Video

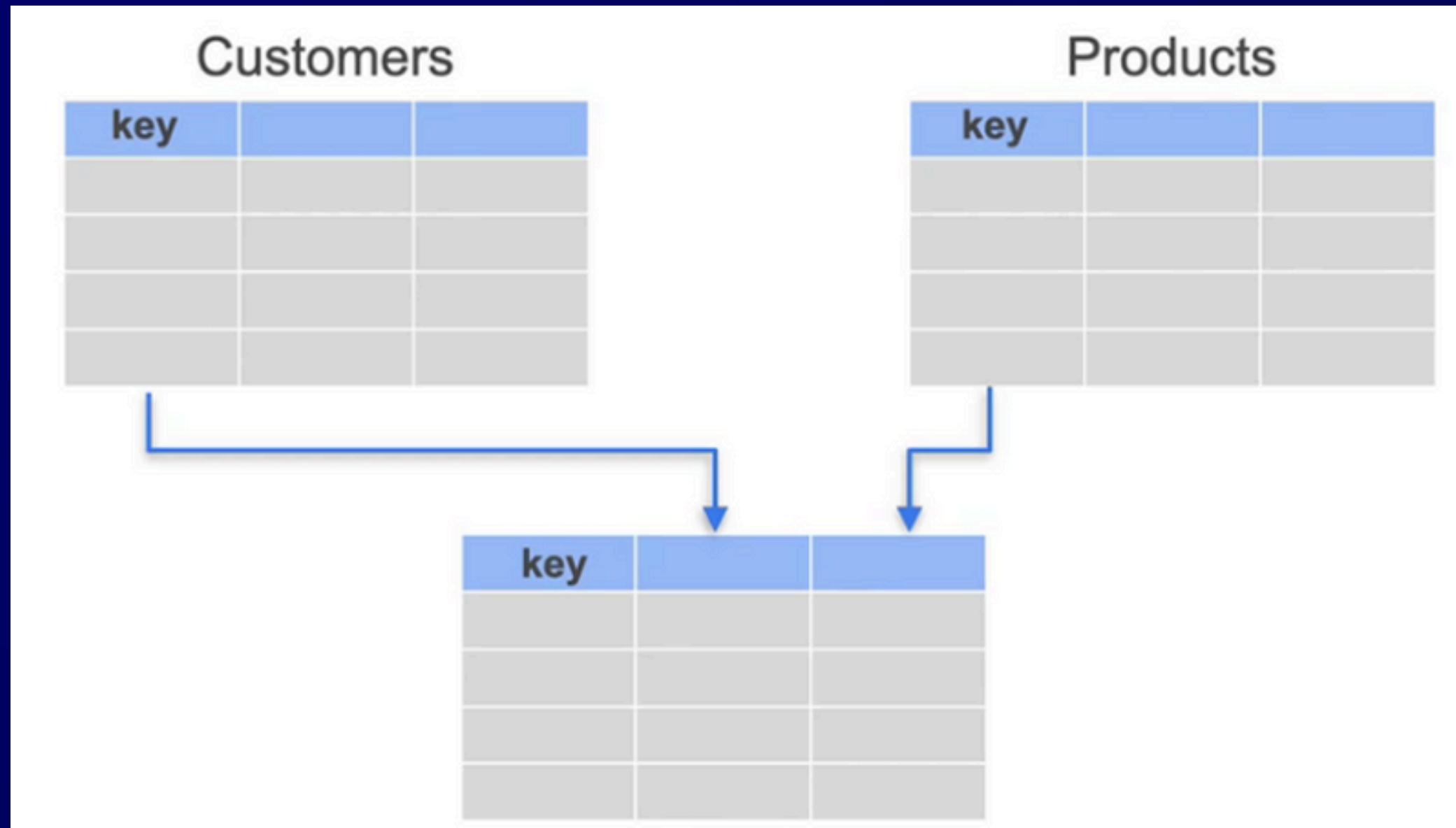


Image

# Relational Databases

---

# Relational Databases



Database Schema

- Giảm thiểu dư thừa
- Làm cho dữ liệu dễ dàng quản lý

# Relational Databases

Một bảng lớn cho tất cả dữ liệu

name	address	phone	date_time	amount	brand	SKU	description
Jane Doe	74th Street	12345678	12/08/2024	700	ABC	B32	Blender
Jane Doe	74th Street	12345678	12/08/2024	99	XYZ	i56	Iron
Jane Doe	74th Street	12345678	12/08/2024	100	GHJ	k70	Kettle
Mary Ann	19th Avenue	98765432	13/08/2024	899	STU	w40	Washer
John Ken	1st Link	36891623	14/08/2024	899	STU	w40	Washer
Ivy Tan	67th Street	98639513	15/08/2024	899	STU	w40	Washer

# Relational Databases

Một bảng lớn cho tất cả dữ liệu

name	address	phone	date_time	amount	brand	SKU	description
Jane Doe	74th Street	12345678	12/08/2024	700	ABC	B32	Blender
Jane Doe	74th Street	12345678	12/08/2024	99	XYZ	i56	Iron
Jane Doe	74th Street	12345678	12/08/2024	100	GHJ	k70	Kettle
Mary Ann	19th Avenue	98765432	13/08/2024	899	STU	w40	Washer
John Ken	1st Link	36891623	14/08/2024	899	STU	w40	Washer
Ivy Tan	67th Street	98639513	15/08/2024	899	STU	w40	Washer

# Relational Databases

## Primary Key

Giá trị độc nhất để  
định danh từng hàng  
trong một bảng

Customers

id	first_name	last_name	age	address
1	Jane	Doe	24	74th St.
2	Mary	Ann	65	19th Ave.
3	John	Ken	27	1st Link
4	Ivy	Tan	18	67th St.

Products

id	brand	SKU	description
1	ABC	b32	Blender
2	XYZ	i56	Iron
3	GHJ	k70	Kettle
4	STU	w40	Washer

Orders

id	customer_id	product_id	date_time	purchase_amount
1	1	1	12/08/2024	700
2	1	2	12/08/2024	99
3	1	3	12/08/2024	100
4	2	4	13/08/2024	899
5	3	4	14/08/2024	899

## Foreign Key

Tham chiếu đến primary key của bảng customers

# Relational Databases

Customers

<b>id</b>	<b>first_name</b>	<b>last_name</b>	<b>age</b>	<b>address</b>
1	Jane	Doe	24	74th St.
2	Mary	Ann	65	19th Ave.
3	John	Ken	27	1st Link
4	Ivy	Tan	18	67th St.

Products

<b>id</b>	<b>brand</b>	<b>SKU</b>	<b>description</b>
1	ABC	b32	Blender
2	XYZ	i56	Iron
3	GHJ	k70	Kettle
4	STU	w40	Washer

Orders

<b>id</b>	<b>customer_id</b>	<b>product_id</b>	<b>date_time</b>	<b>purchase_amount</b>
1	1	1	12/08/2024	700
2	1	2	12/08/2024	99
3	1	3	12/08/2024	100
4	2	4	13/08/2024	899
5	3	4	14/08/2024	899

Mỗi hàng trong bảng đều có cùng cấu trúc cột: cùng số lượng cột và kiểu dữ liệu

# NoSQL Databases

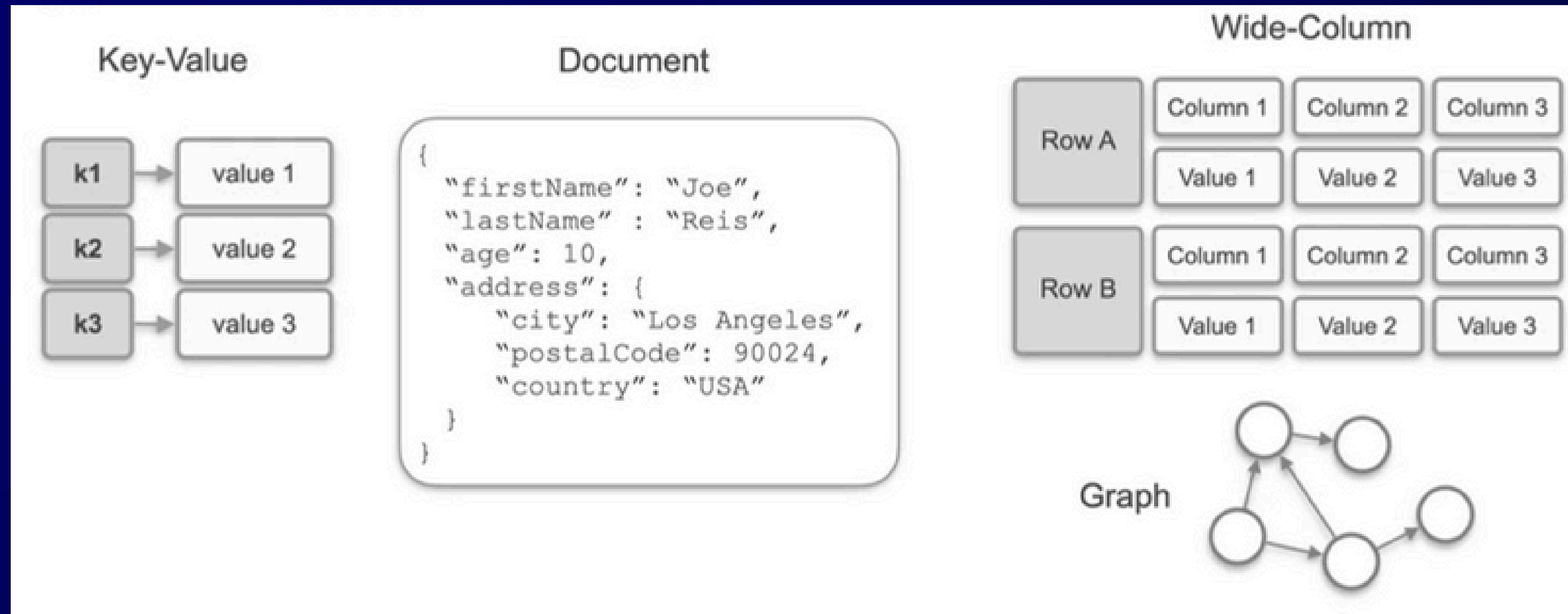
Not Only SQL

Non-Relational Databases

Vẫn có thể hỗ trợ SQL hoặc những ngôn ngữ giống SQL

---

# NoSQL Databases



- Không cần định nghĩa trước schema
- Linh hoạt hơn trong việc lưu trữ dữ liệu

# Document Database



# Database ACID Compliance

# Database ACID Compliance

## Relational Databases

### ACID

Atomicity - Tính nguyên tử

Consistency - Tính nhất quán

Isolation - Tính cô lập

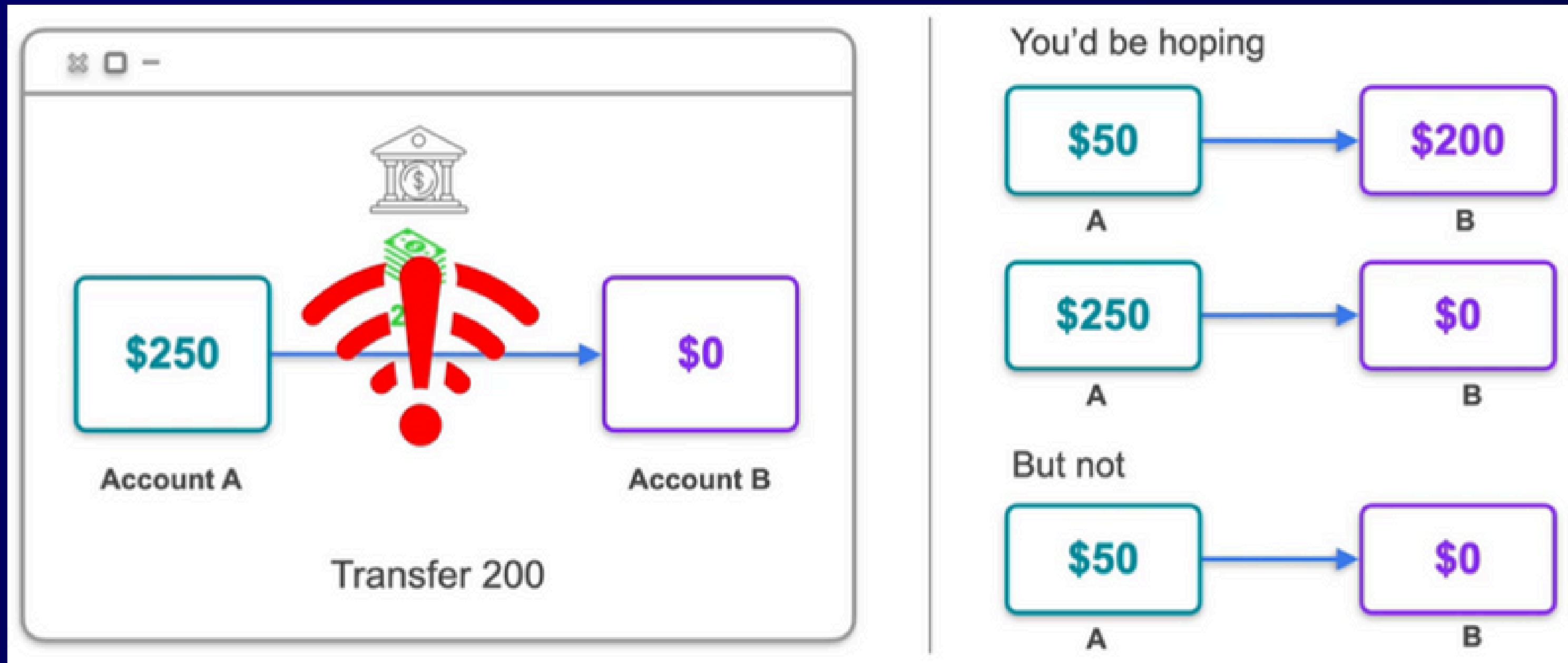
Durability - Tính bền vững

Việc tuân thủ ACID đảm bảo các giao dịch được xử lý với độ tin cậy và độ chính xác cao trong hệ thống OLTP

## NoSQL Databases

Thường mặc định không tuân thủ ACID

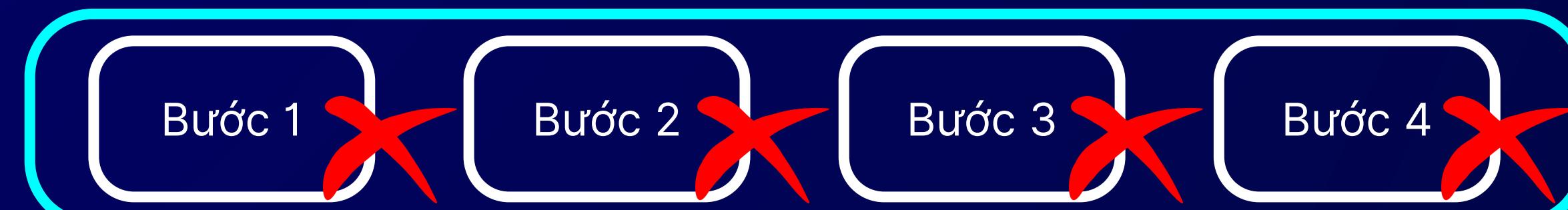
# Database ACID Compliance



## Atomicity

Tính nguyên tử giúp đảm bảo giao dịch là **nguyên tử**, sao cho tất cả các bước đều xảy ra hoặc là không gì xảy ra.

### Một giao dịch



## Atomicity

Tính nguyên tử giúp đảm bảo giao dịch là **nguyên tử**, sao cho tất cả các bước đều xảy ra hoặc là không gì xảy ra.

### Một giao dịch chuyển tiền

Trừ đi số tiền trong tài khoản người chuyển



Thêm số tiền vào tài khoản người nhận



Cả hai hành động đều phải xảy ra như một giao dịch đơn lẻ

Trừ đi số tiền trong tài khoản người chuyển



Thêm số tiền vào tài khoản người nhận



Tài khoản của người chuyển được ROLLBACK.

## Atomicity

Tính nguyên tử giúp đảm bảo giao dịch là **nguyên tử**, là một đơn vị riêng lẻ, sao cho tất cả các bước đều xảy ra hoặc là không gì xảy ra.

## Consistency

Mỗi thay đổi về dữ liệu trong một giao dịch phải tuân theo quy tắc hoặc những bắt buộc được định nghĩa bởi database schema.



## Atomicity

Tính nguyên tử giúp đảm bảo giao dịch là **nguyên tử**, là một đơn vị riêng lẻ, sao cho tất cả các bước đều xảy ra hoặc là không gì xảy ra.

## Consistency

Mỗi thay đổi về dữ liệu trong một giao dịch phải tuân theo quy tắc hoặc những bắt buộc được định nghĩa bởi database schema.

## Isolation

Tính cô lập đảm bảo các transaction được thực thi một cách độc lập, không phụ thuộc lẫn nhau.

id	product_name	quantity
1	iPhone 16	10

Giao dịch A

Mua 5 iPhone 16

Giao dịch B

Mua 10 iPhone 16



## Atomicity

Tính nguyên tử giúp đảm bảo giao dịch là **nguyên tử**, là một đơn vị riêng lẻ, sao cho tất cả các bước đều xảy ra hoặc là không gì xảy ra.

## Consistency

Mỗi thay đổi về dữ liệu trong một giao dịch phải tuân theo quy tắc hoặc những bắt buộc được định nghĩa bởi database schema.

## Isolation

Tính cô lập đảm bảo các transaction được thực thi một cách độc lập, không phụ thuộc lẫn nhau.

## Durability

Tính bền vững đảm bảo rằng một khi một giao dịch đã được cam kết (commit) thành công, các thay đổi của nó sẽ là vĩnh viễn và không bị mất ngay cả khi có lỗi hệ thống, mất điện, hoặc các sự cố khác xảy ra sau đó.

### Một giao dịch chuyển tiền

Trừ đi số tiền trong tài khoản người chuyển

Thêm số tiền vào tài khoản người nhận

## Atomicity

Tính nguyên tử giúp đảm bảo giao dịch là **nguyên tử**, là một đơn vị riêng lẻ, sao cho tất cả các bước đều xảy ra hoặc là không gì xảy ra.

## Consistency

Mỗi thay đổi về dữ liệu trong một giao dịch phải tuân theo quy tắc hoặc những bắt buộc được định nghĩa bởi database schema.

## Isolation

Tính cô lập đảm bảo các transaction được thực thi một cách độc lập, không phụ thuộc lẫn nhau.

## Durability

Tính bền vững đảm bảo rằng một khi một giao dịch đã được cam kết (commit) thành công, các thay đổi của nó sẽ là vĩnh viễn và không bị mất ngay cả khi có lỗi hệ thống, mất điện, hoặc các sự cố khác xảy ra sau đó.

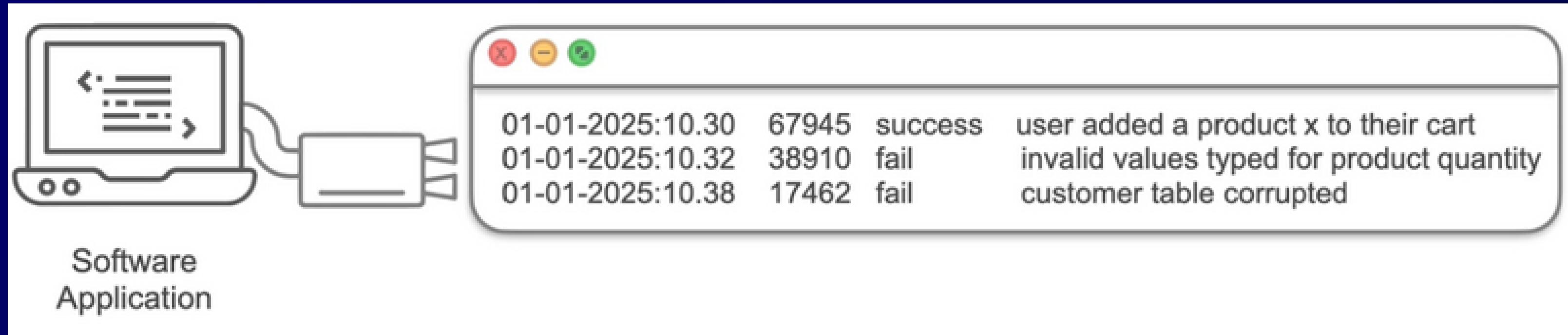
### Một giao dịch chuyển tiền

Trừ đi số tiền trong tài khoản người chuyển

Thêm số tiền vào tài khoản người nhận

# Logs

---



## Giám sát hoặc Gỡ lỗi hệ thống

- Hoạt động người dùng
  - Đăng nhập
  - Chuyển đến một trang cụ thể
- Một cập nhật vào trong database
- Lỗi từ một quy trình

# Log

Một bản ghi lại các thông tin sự kiện xuất hiện trong hệ thống theo thời gian.



## Log

Một bản ghi lại các thông tin sự kiện xuất hiện trong hệ thống theo thời gian.

### Một nhãn để phân loại các sự kiện (log level)

- DEBUG
- INFO
- WARN
- ERROR
- FATAL

user id	action	status	timestamp	level
67945	user added a product x to their cart	success	01-01-2025:10.30	Info
38910	invalid values typed for product quantity	fail	01-01-2025:10.32	error
17462	customer table corrupted	fail	01-01-2025:10.38	fatal

# Streaming Systems

---

# Các thuật ngữ

## Event

Một sự kiện hoặc hành động đã xảy ra, hoặc thay đổi trạng thái hệ thống



Người dùng click vào một link

## Message

Một bản ghi về thông tin của “event”

### Message

Event Details  
Event Metadata  
Event Timestamp

## Stream

Một chuỗi các “message”

## Producer

## Event

## Event

## Event

## Event

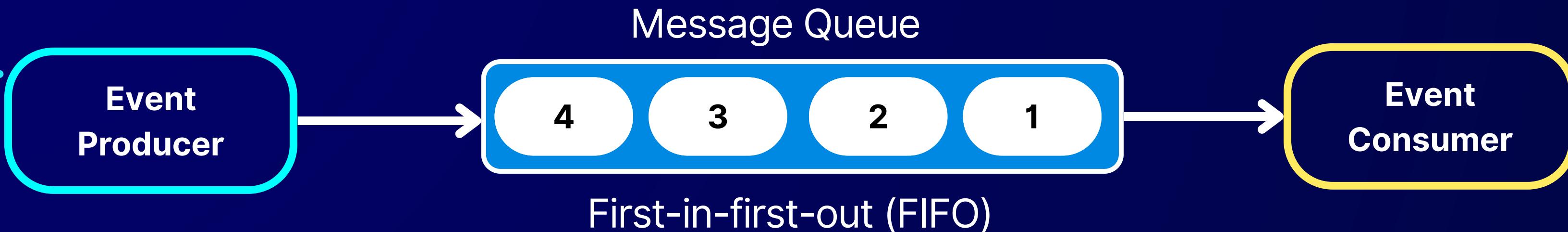
← Stream



# Các loại streaming system chính

## Message Queue

Một hàng đợi tích lũy messages và gửi chúng đến người nhận không đồng bộ.



### Đặc điểm

- Giao tiếp không đồng bộ: Producer và consumer không cần hoạt động cùng lúc, giúp giảm sự phụ thuộc giữa các thành phần.
- Lưu trữ tạm thời, chủ yếu để chờ xử lý.
- Khi message đã được tiêu thụ thành công, nó sẽ bị xóa khỏi queue.

### Ứng dụng

- Xử lý tác vụ nền, công việc lâu, gửi email
- Xử lý đơn hàng, thanh toán, thông báo
- Kết nối microservices, giảm tải hệ thống
- Đảm bảo dữ liệu không mất khi lỗi dịch vụ
- ...

# Các loại streaming system chính

## Event Streaming Platform

Lưu trữ các event (message) thành các log bất biến (immutable log).

### Event Producer

4

3

2

1

### Event Consumer

### Đặc điểm

- Lưu trữ lâu dài, cho phép nhiều consumer đọc cùng một event ở các thời điểm khác nhau
- Event không bị xóa ngay sau khi tiêu thụ.
- Hỗ trợ replay event (đọc lại event cũ)

### Ứng dụng

- Phân tích dữ liệu real-time, phát hiện gian lận
  - Theo dõi sensor, IoT, tối ưu sản xuất
  - Cá nhân hóa nội dung, phân tích hành vi người dùng
  - Quản lý logistics, tối ưu vận tải
  - Xây dựng data pipeline, tích hợp đa nguồn dữ liệu

# So sánh các loại streaming system

## So sánh Message Queue và Event Streaming Platform

Tiêu chí	Message Queue	Event Streaming Platform
Mục đích	Phân phối công việc, đảm bảo xử lý 1 lần	Xử lý dữ liệu real-time, analytics, event sourcing
Mô hình giao tiếp	Point-to-point (1 producer - 1 consumer)	Publish-Subscribe (1 producer - nhiều consumer)
Tiêu thụ dữ liệu	Xóa sau khi consumer xử lý xong	Lưu trữ sự kiện, cho phép replay
Độ trễ	Thấp, phù hợp task queue	Rất thấp, tối ưu cho real-time
Khả năng mở rộng	Hạn chế hơn, thường theo chiều dọc	Rất tốt, phân tán, mở rộng theo chiều ngang
Ứng dụng tiêu biểu	Task queue, workload distribution	Real-time analytics, log aggregation, IoT
Ví dụ	RabbitMQ, ActiveMQ, Amazon SQS	Apache Kafka, AWS Kinesis, Google Pub/Sub

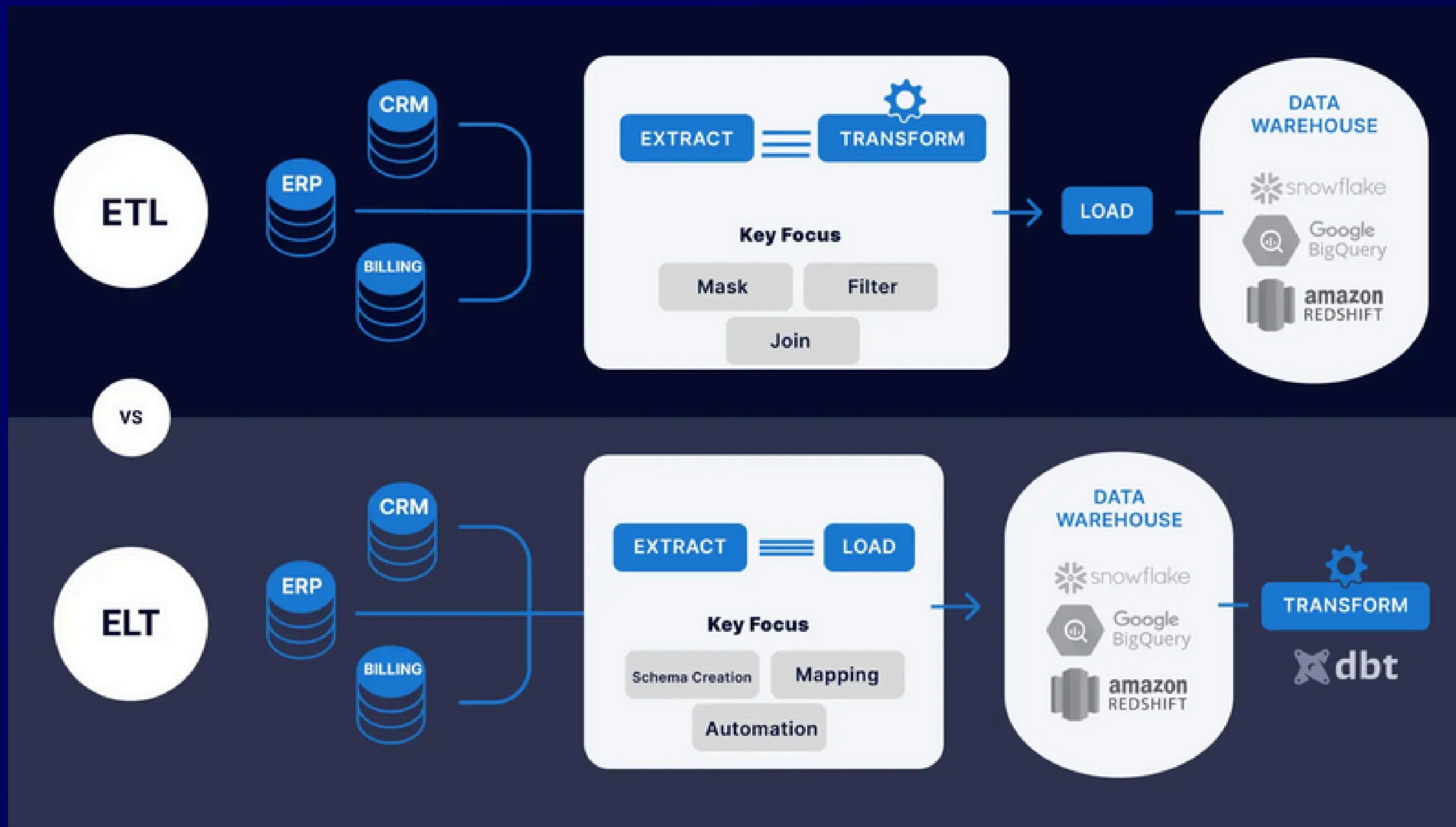
# DATA INGESTION

---

# Batch Ingestion

---

# ETL vs ELT



## Tóm tắt các điểm khác biệt giữa ETL và ELT

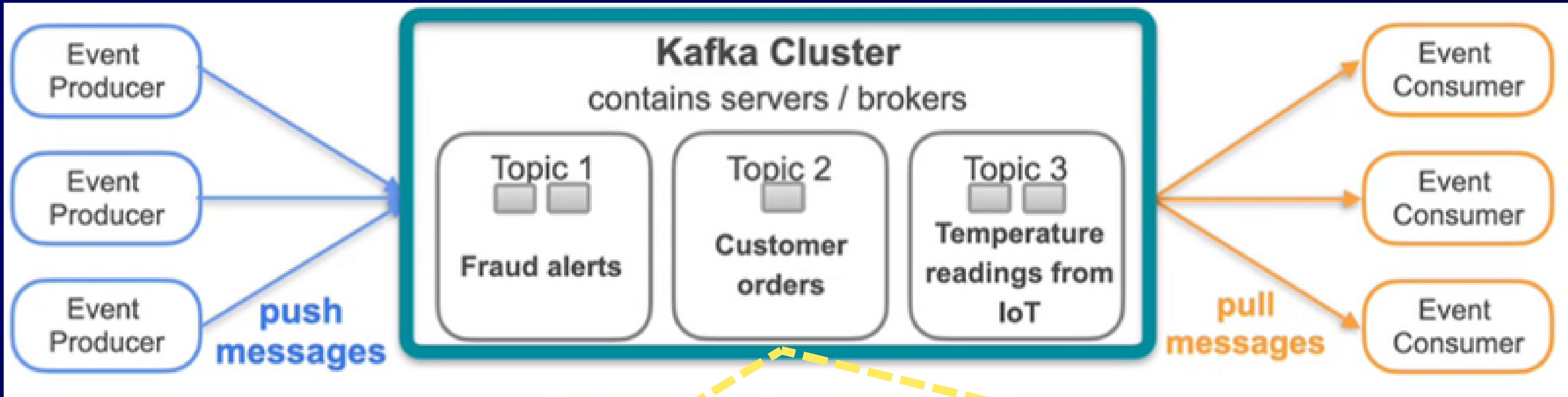
Danh mục	ETL	ELT
Là viết tắt của	Trích xuất, chuyển đổi và tải	Trích xuất, tải và chuyển đổi
Quy trình	Lấy dữ liệu thô và chuyển đổi dữ liệu thô thành một định dạng đã xác định trước, sau đó tải vào kho dữ liệu mục tiêu.	Lấy dữ liệu thô, tải dữ liệu thô vào kho dữ liệu mục tiêu, sau đó chuyển đổi dữ liệu thô ngay trước khi phân tích.
Vị trí chuyển đổi và tải	Quá trình chuyển đổi diễn ra trong một máy chủ xử lý thứ cấp.	Quá trình chuyển đổi diễn ra trong kho dữ liệu mục tiêu.
Khả năng tương thích với dữ liệu	Phù hợp nhất với dữ liệu có cấu trúc.	Có thể xử lý dữ liệu có cấu trúc, phi cấu trúc và bán cấu trúc.
Tốc độ	ETL chậm hơn ELT.	ELT nhanh hơn ETL vì nó có thể sử dụng các tài nguyên nội bộ của kho dữ liệu.
Chi phí	Có thể cần nhiều thời gian và chi phí để thiết lập, tùy thuộc vào các công cụ ETL được sử dụng.	Tiết kiệm chi phí hơn, tùy thuộc vào cơ sở hạ tầng ELT được sử dụng.
Bảo mật	Có thể cần phải xây dựng các ứng dụng tùy chỉnh để đáp ứng các yêu cầu bảo vệ dữ liệu.	Bạn có thể sử dụng các tính năng tích hợp sẵn của cơ sở dữ liệu mục tiêu để quản lý việc bảo vệ dữ liệu.

# Streaming Ingestion

---



Kho dữ liệu phân tán này được tối ưu để thu nạp và xử lý liên tục dữ liệu truyền phát theo thời gian thực từ nhiều nguồn, đảm bảo xử lý tuần tự và tăng dần với độ trễ thấp.



### Topics:

Phân loại và lưu trữ các event liên quan đến nhau

### Partitions (logs):

Chuỗi messages liên tiếp được sắp xếp

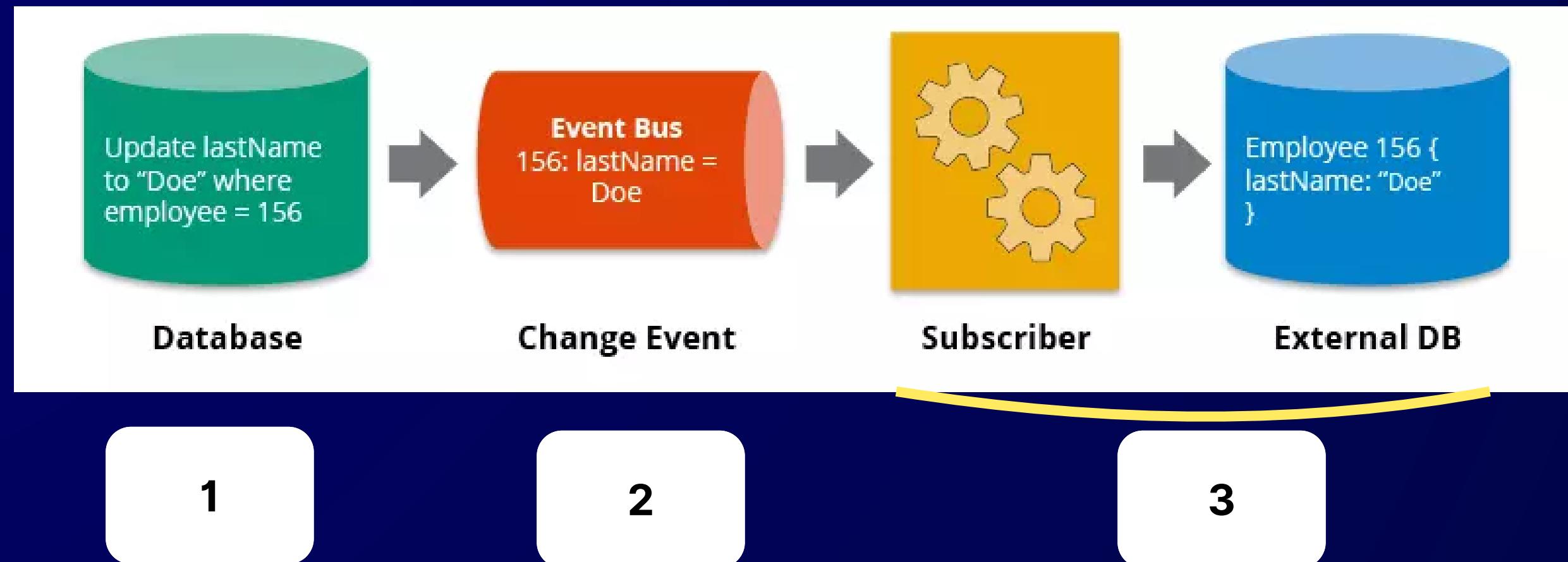


### Partition decision:

- Chiến lược Round-robin
- Message Key

# Change Data Capture (CDC)

Đúng như cái tên của nó là **bắt sự thay đổi dữ liệu**, đây là kỹ thuật sử dụng để chúng ta bắt được những sự thay đổi về dữ liệu chứa trong database



**Phát hiện thay đổi:**  
CDC liên tục theo dõi các sự kiện thay đổi trên dữ liệu nguồn (INSERT, UPDATE, DELETE).

**Ghi nhận thay đổi:**  
Khi phát hiện thay đổi, CDC sẽ ghi lại thông tin liên quan như loại thao tác, thời điểm xảy ra, dữ liệu cũ và dữ liệu mới (nếu có).

**Truyền tải thay đổi:**  
Các thay đổi này được chuyển tiếp đến hệ thống đích, đảm bảo dữ liệu luôn được cập nhật mới nhất mà không cần tải lại toàn bộ bảng dữ liệu.

# DATA STORAGE

---

# OLTP vs OLAP

---

Tiêu chí	OLAP	OLTP
Mục đích	OLAP giúp bạn phân tích khối lượng dữ liệu lớn để hỗ trợ quá trình ra quyết định.	OLTP giúp bạn quản lý và xử lý các giao dịch thời gian thực.
Nguồn dữ liệu	OLAP sử dụng dữ liệu lịch sử và dữ liệu tổng hợp từ nhiều nguồn.	OLTP sử dụng dữ liệu thời gian thực và dữ liệu giao dịch từ một nguồn duy nhất.
Cấu trúc dữ liệu	OLAP sử dụng cơ sở dữ liệu đa chiều (khối) hoặc cơ sở dữ liệu quan hệ.	OLTP sử dụng cơ sở dữ liệu quan hệ.
Mô hình dữ liệu	OLAP sử dụng lược đồ ngôi sao, lược đồ bông tuyết, hoặc các mô hình phân tích khác.	OLTP sử dụng các mô hình chuẩn hóa hoặc phi chuẩn hóa.
Khối lượng dữ liệu	OLAP có yêu cầu lưu trữ lớn. Hãy nghĩ đến hàng terabyte (TB) và hàng petabyte (PB).	OLTP có yêu cầu lưu trữ tương đối nhỏ hơn. Hãy nghĩ đến hàng gigabyte (GB).
Thời gian phản hồi	OLAP có thời gian phản hồi dài hơn, thông thường tính bằng giây hoặc phút.	OLTP có thời gian phản hồi ngắn hơn, thường tính bằng mili giây
Ví dụ về trường hợp ứng dụng	OLAP rất phù hợp với việc phân tích xu hướng, dự đoán hành vi của khách hàng và xác định khả năng sinh lời.	OLTP rất phù hợp với việc xử lý thanh toán, quản lý dữ liệu khách hàng và xử lý đơn hàng.

# Row vs Column Storage

---

# Row-Oriented Storage

Order ID	Price	Product SKU	Quantity	Customer ID
1	40	45865	10	67t
2	23	90234	14	56t
3	45	12558	12	87q
4	50	45682	13	98q

**Row Storage hoàn hảo  
cho OLTP**

Thực hiện đọc và ghi dữ liệu với  
độ trễ thấp

Dữ liệu lưu trữ thành  
từng hàng

Lưu trữ vật lý

1	40	45865	10	67t	2	23	90234	14	56t	...	4	50	45682	13	98q
---	----	-------	----	-----	---	----	-------	----	-----	-----	---	----	-------	----	-----

# Column-Oriented Storage

Order ID	Price	Product SKU	Quantity	Customer ID
1	40	45865	10	67t
2	23	90234	14	56t
3	45	12558	12	87q
4	50	45682	13	98q

Column Storage hoàn  
hảo cho OLAP

Không phù hợp với các khối  
lượng công việc giao dịch

↓  
Dữ liệu được lưu trữ  
theo từng cột

Lưu trữ vật lý

1	2	3	4	40	23	45	50	45865	90234	12558	45682	...
---	---	---	---	----	----	----	----	-------	-------	-------	-------	-----

# Data Warehouse

---

# Data Warehouse

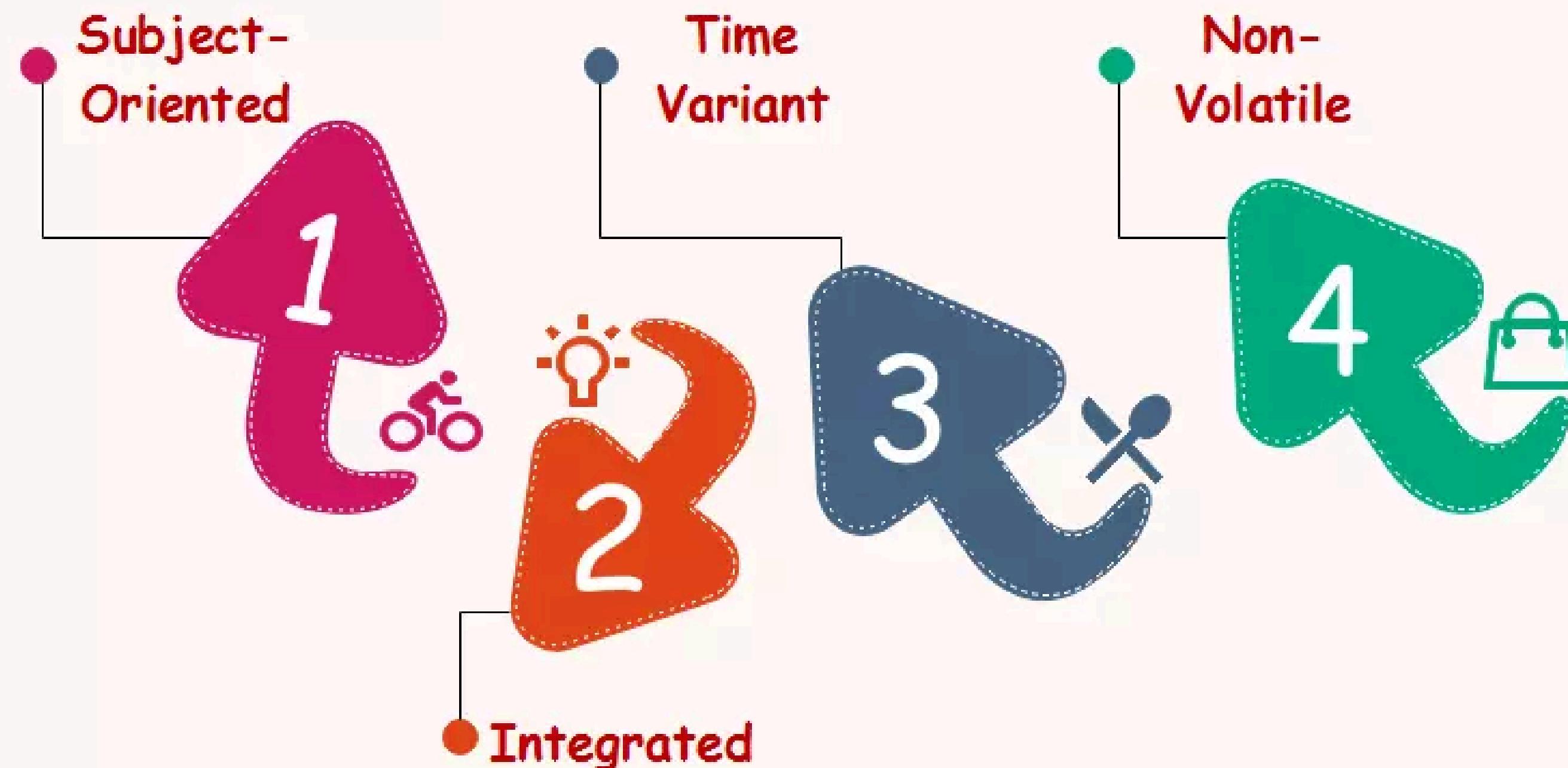


Là hệ thống lưu trữ dữ liệu tập trung, được thiết kế để tổng hợp và phân tích dữ liệu từ nhiều nguồn khác nhau nhằm hỗ trợ quá trình ra quyết định trong doanh nghiệp.

Khác với dữ liệu giao dịch (OLTP), DW tập trung vào việc truy vấn, phân tích dữ liệu lịch sử và cung cấp thông tin tổng quan đa chiều cho các nhà phân tích, quản lý

# Đặc điểm cơ bản của Data Warehouse

The key features of Data Warehouse are:

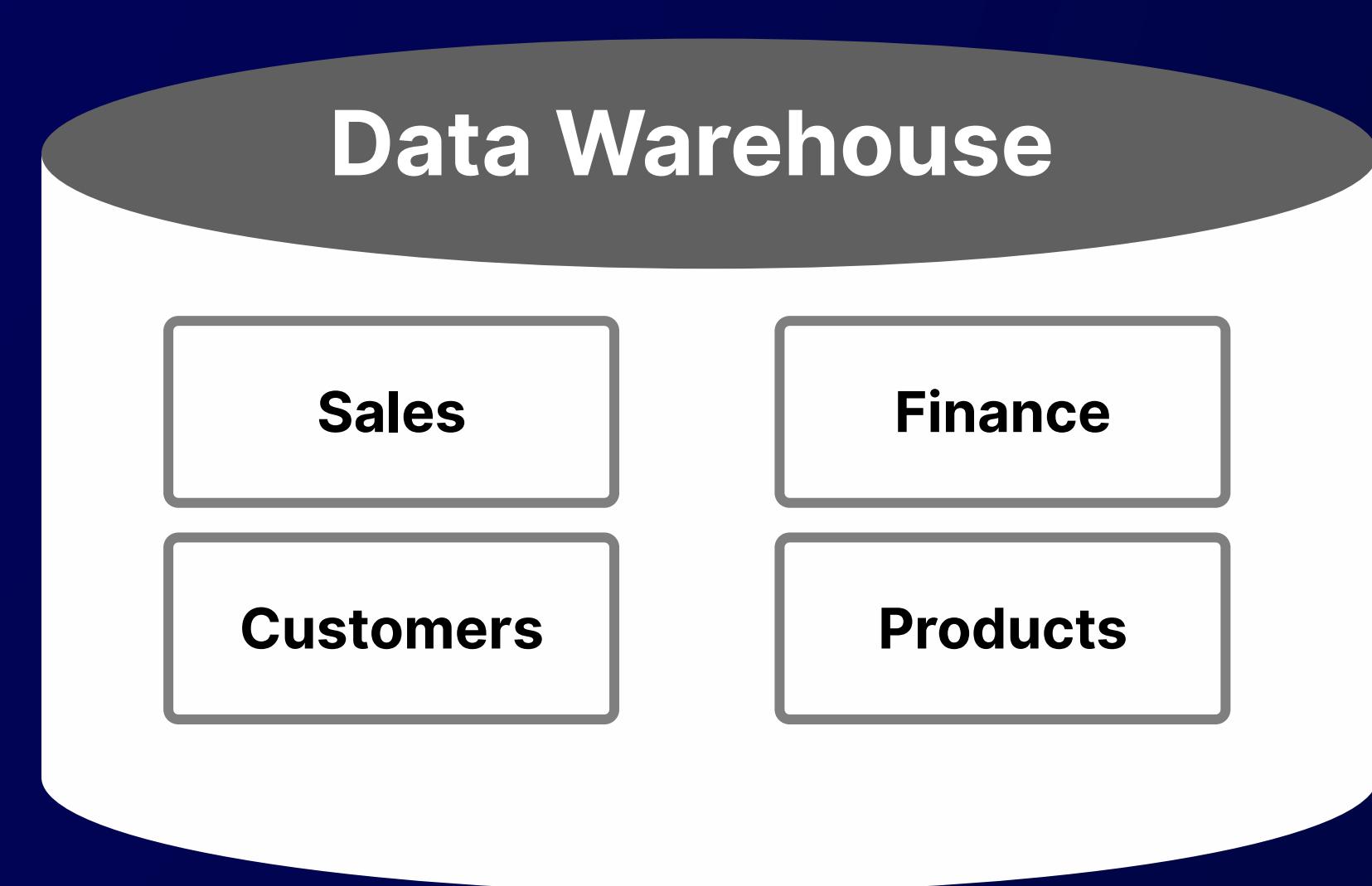


# Đặc điểm cơ bản của Data Warehouse

## Subject-Oriented

### Hướng chủ đề

Tổ chức và lưu trữ xoay quanh các chủ đề lớn như bán hàng, tài chính, khách hàng... thay vì quy trình nghiệp vụ cụ thể

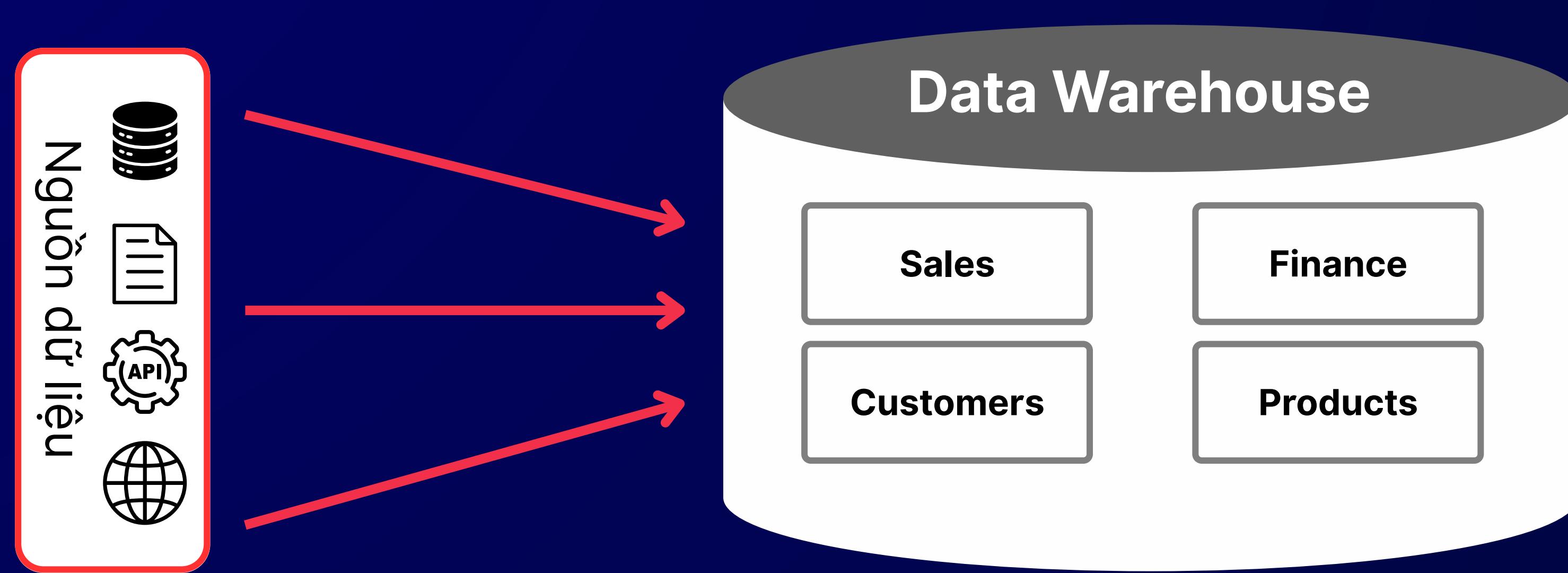


# Đặc điểm cơ bản của Data Warehouse

Integrated

Tích hợp

Kết hợp dữ liệu từ nhiều nguồn vào một định dạng nhất quán

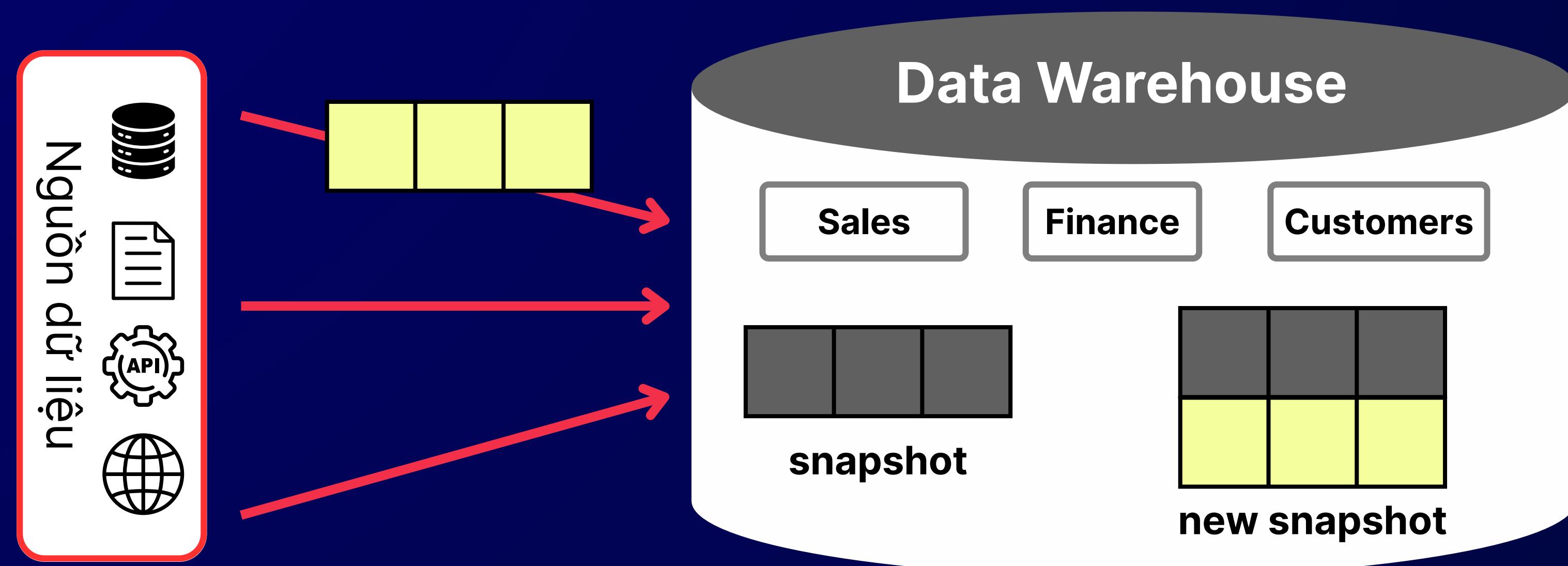


# Đặc điểm cơ bản của Data Warehouse

Nonvolatile

Bất biến

Dữ liệu là read-only (chỉ đọc) và không thể xóa hoặc cập nhật

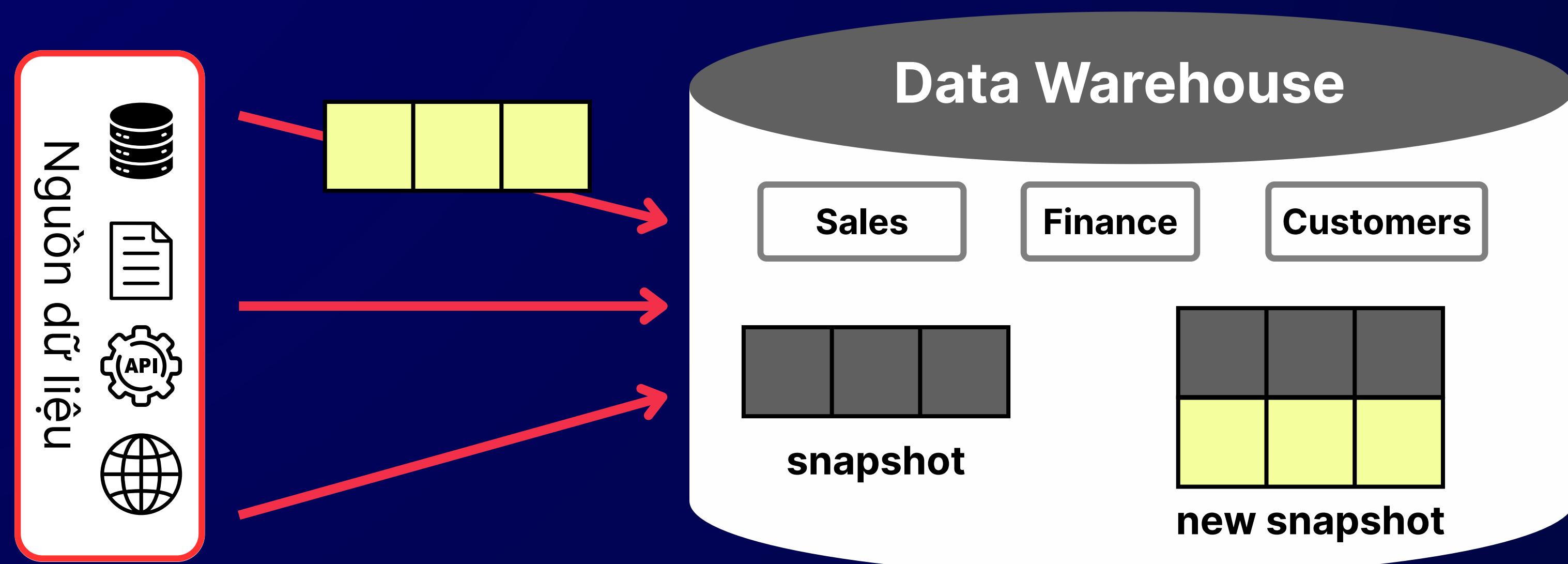


# Đặc điểm cơ bản của Data Warehouse

## Time-variant

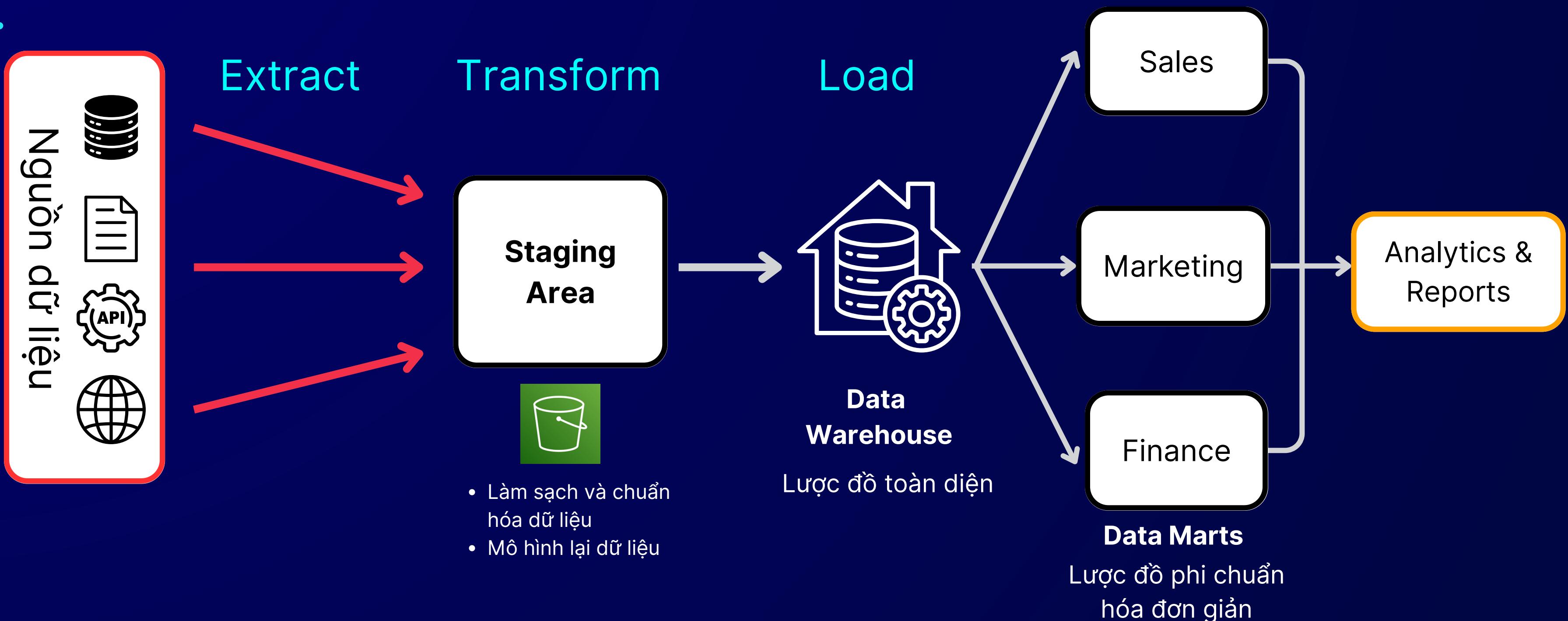
### Biến đổi theo thời gian

Lưu trữ dữ liệu hiện tại và dữ liệu lịch sử, cho phép truy xuất lịch sử và so sánh dữ liệu theo từng giao đoạn



# Kiến trúc Data Warehouse tập trung

## Extract-Transform-Load (ETL)



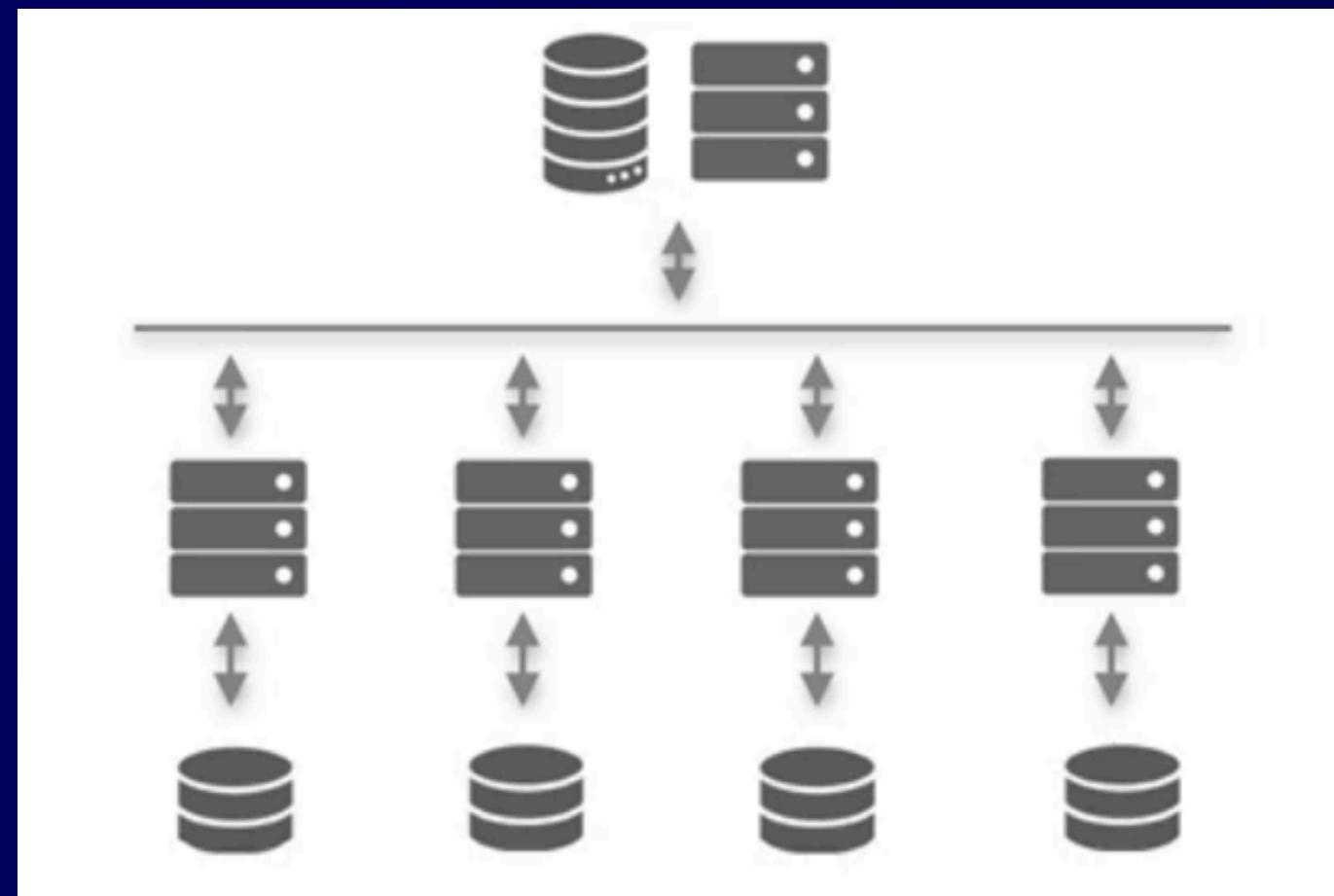
# Triển khai Data Warehouse

Data Warehouses  
ban đầu



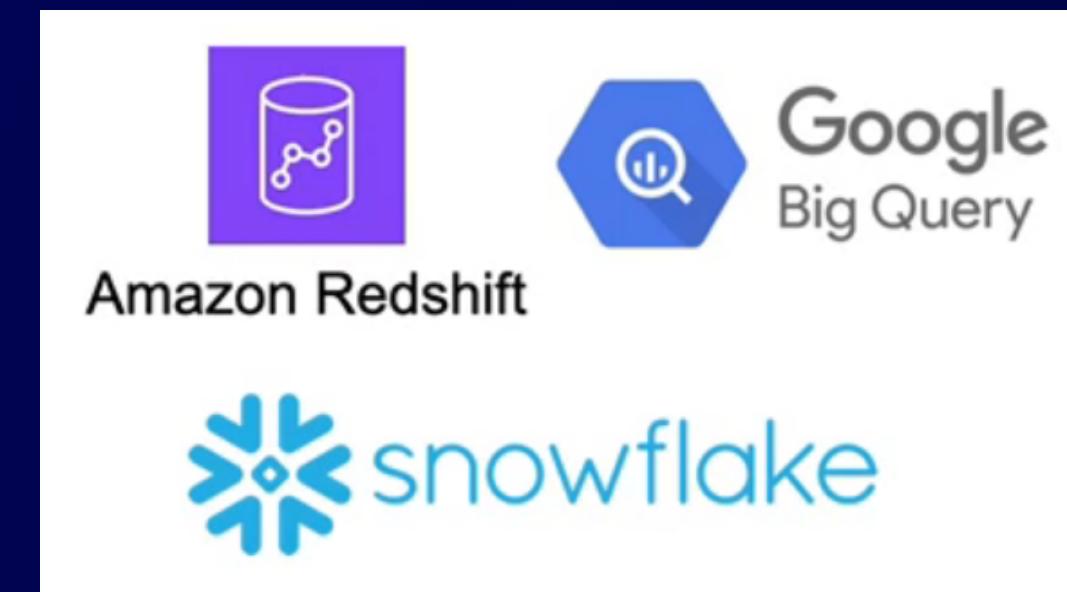
Big monolithic  
server

Data Warehouses với  
Massively Parallel Processing  
(MPP)



- Quét lượng lớn dữ liệu song song
- Cấu hình phức tạp và cần nhiều công sức để duy trì

Modern Cloud  
Data Warehouses

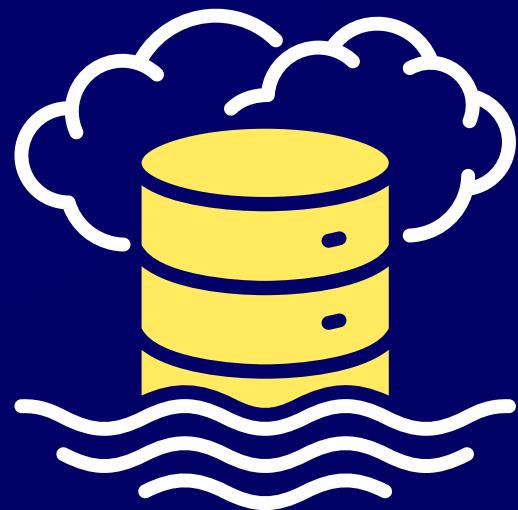


- Tách biệt tính toán và lưu trữ
- Mở rộng khả năng của hệ thống MPP

# Data Lake

---

# Data Lake



- Là kho lưu trữ trung tâm cho khối lượng lớn dữ liệu
- Không yêu cầu lược đồ (schema) cố định hoặc định nghĩa trước các transformations
- **Schema-on-read pattern:**
  - Người đọc dữ liệu xác định schema khi đọc dữ liệu

## Data Lake 1.0

Kết hợp công nghệ lưu trữ và tính toán riêng biệt



Storage



Processing  
Tools



Apache Pig



# Những thiếu sót của Data Lake 1.0



## Data Swamp (Đầm lầy dữ liệu)

- Thiếu quản lý, thiếu metadata
- Không có quy trình quản trị
- Dữ liệu hỗn loạn, khó tìm kiếm

## Write-only storage

- Ban đầu chỉ hỗ trợ ghi dữ liệu, các thao tác chỉnh sửa dữ liệu rất khó thực hiện
- Khó để tuân thủ các quy định về dữ liệu

## Không quản lý lược đồ và mô hình dữ liệu

- Khó xử lý dữ liệu lưu trữ
- Dữ liệu không tối ưu cho các truy vấn như JOINs.

# Data Lake

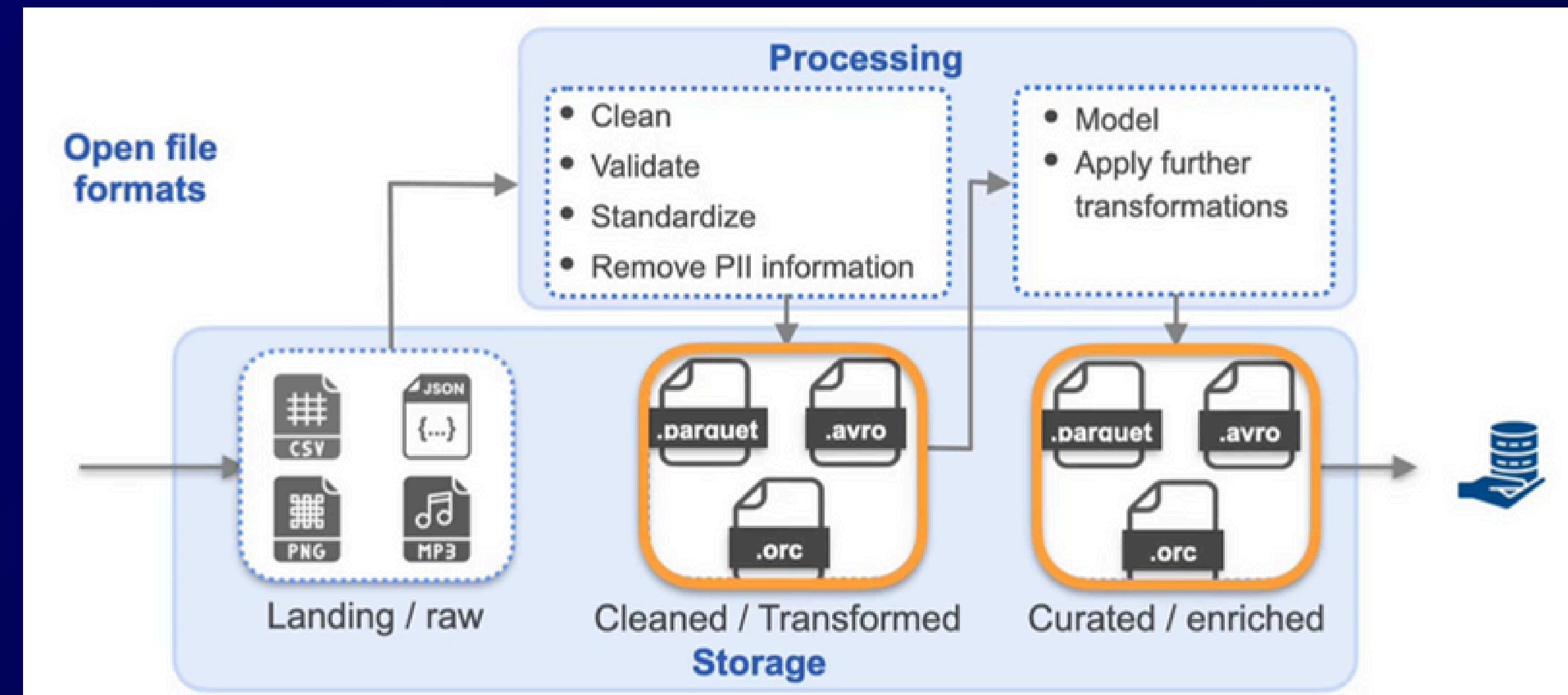
(Thế hệ tiếp theo)



## Data Zones

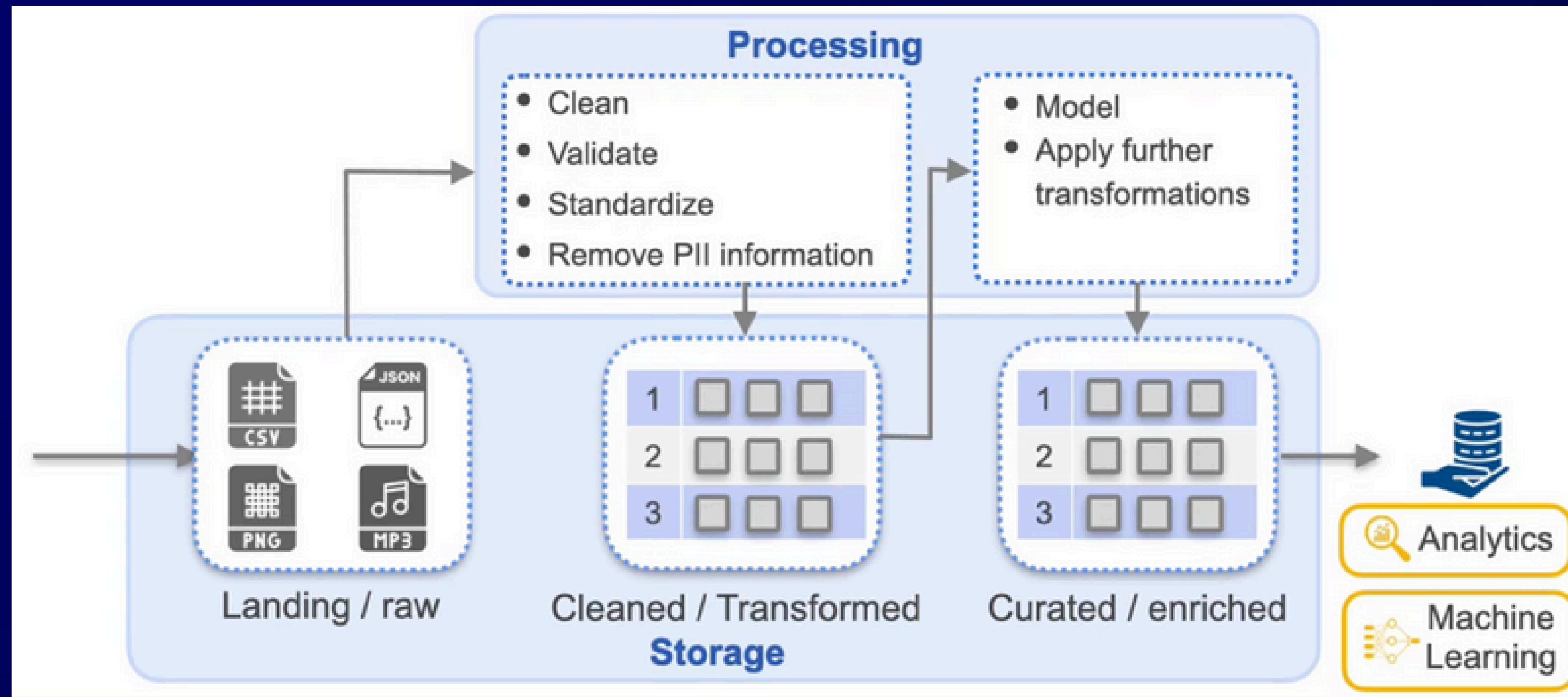
Được sử dụng để tổ chức dữ liệu trong Data Lake, mỗi zone lưu trữ dữ liệu được lưu trữ dữ liệu được xử lý ở mức độ khác nhau

- Áp dụng những chính sách thích hợp quản trị cho từng zone
- Đảm bảo chất lượng dữ liệu



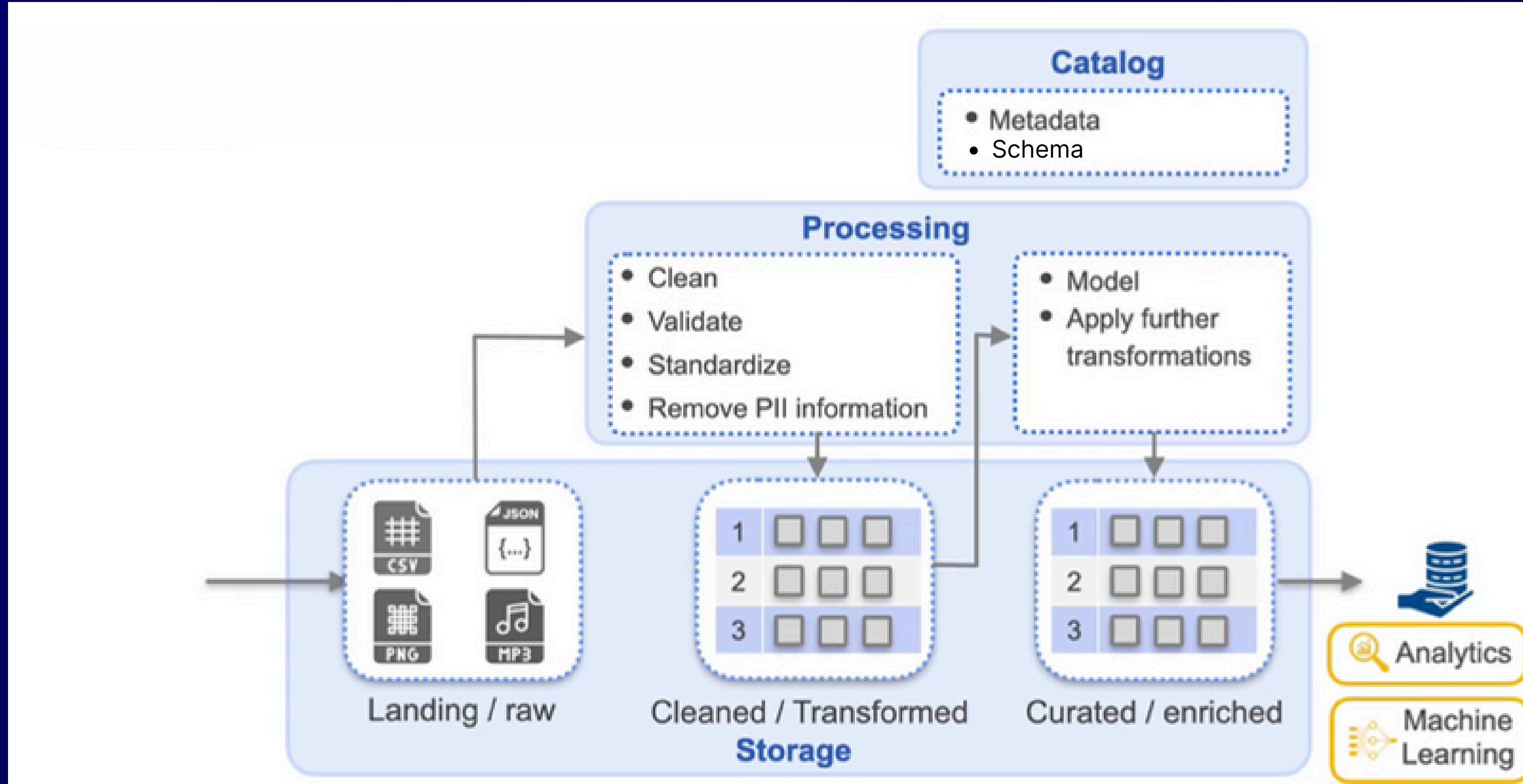
## Data Partitioning

Chia dữ liệu thành nhiều phần nhỏ hơn, dễ quản lý hơn dựa trên một tập các tiêu chuẩn (thời gian, địa điểm được lưu lại trong dữ liệu)



## Data Catalog

Tập các metadata về tập dữ liệu (như chủ sở hữu, nguồn, phân vùng...).  
Đóng vai trò như "bản đồ" của dữ liệu, cho biết dữ liệu nào đang có, nằm ở đâu, chất lượng ra sao, nguồn gốc và mối liên hệ với các dữ liệu khác



# Data Lakehouse

---

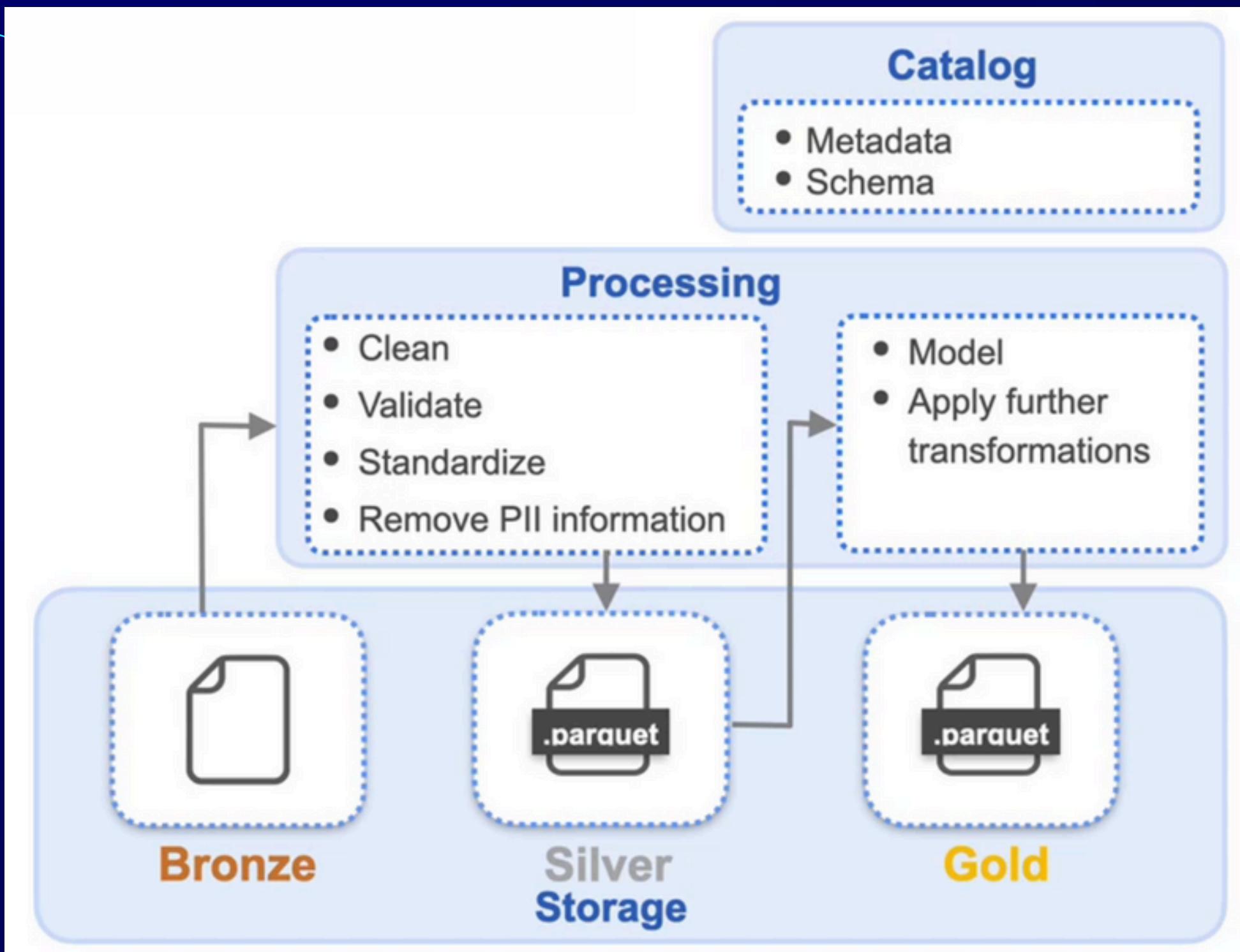
# Data Lakehouse



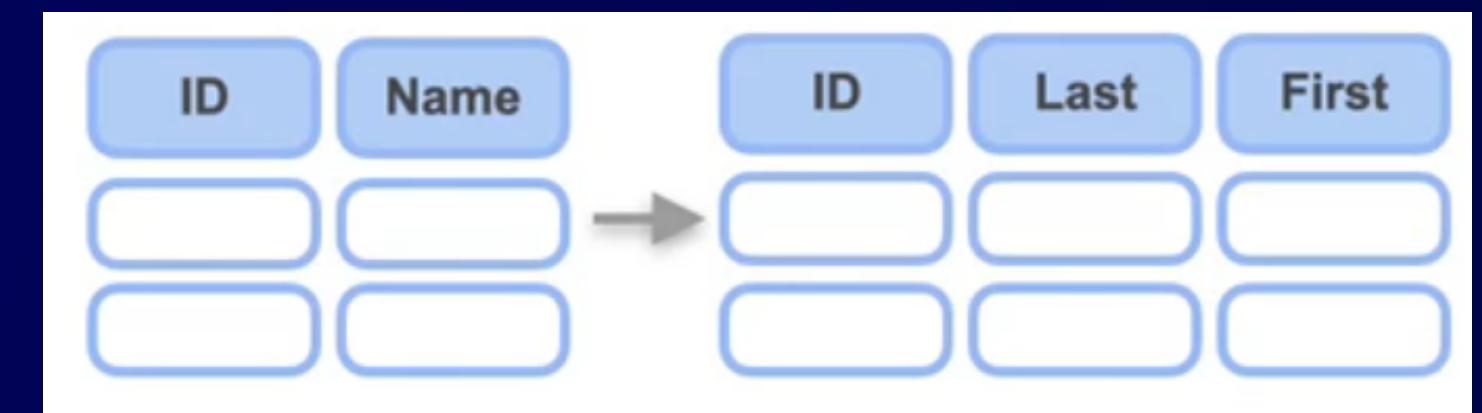
- Hiệu suất truy vấn vượt trội
- Khả năng quản lý dữ liệu mạnh mẽ
- Linh hoạt
- Chi phí lưu trữ thấp



## Medallion Architecture



## Khả năng quản lý dữ liệu từ Data Warehouse Schema Enforcement



### ACID

- Atomic, Consistent, Isolated, Durable
- Khả năng read, update, insert, delete đồng thời

### Data Governance & Security

- Khả năng kiểm soát truy cập mạnh mẽ, kiểm tra dữ liệu, data lineage.
- Liên tục Insert & Deletions
- Quay lại bất kỳ phiên bản dữ liệu lịch sử nào

# Open Table Formats

## Open Table Formats

Định dạng lưu trữ đặc biệt mang lại khả năng thêm giao dịch (transactions) cho data lakehouse.

- Cho phép cập nhật và xóa các bản ghi.
- Hỗ trợ ACID

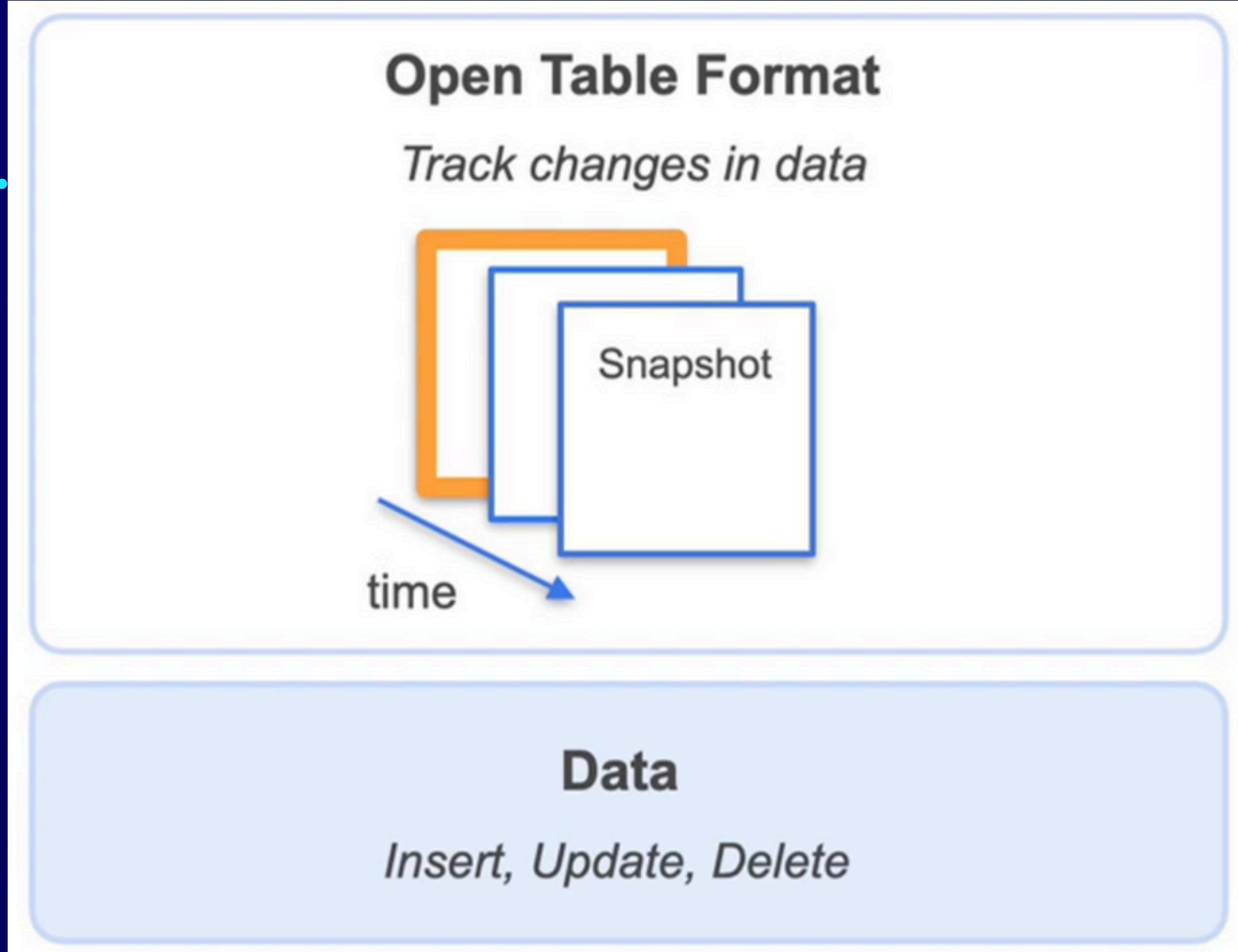


ICEBERG



- Hadoop
- Update
- Delete
- Incremental

# Open Table Formats



## **Snapshot:**

Phản ánh trạng thái dữ liệu tại một thời điểm

## **Time travel:**

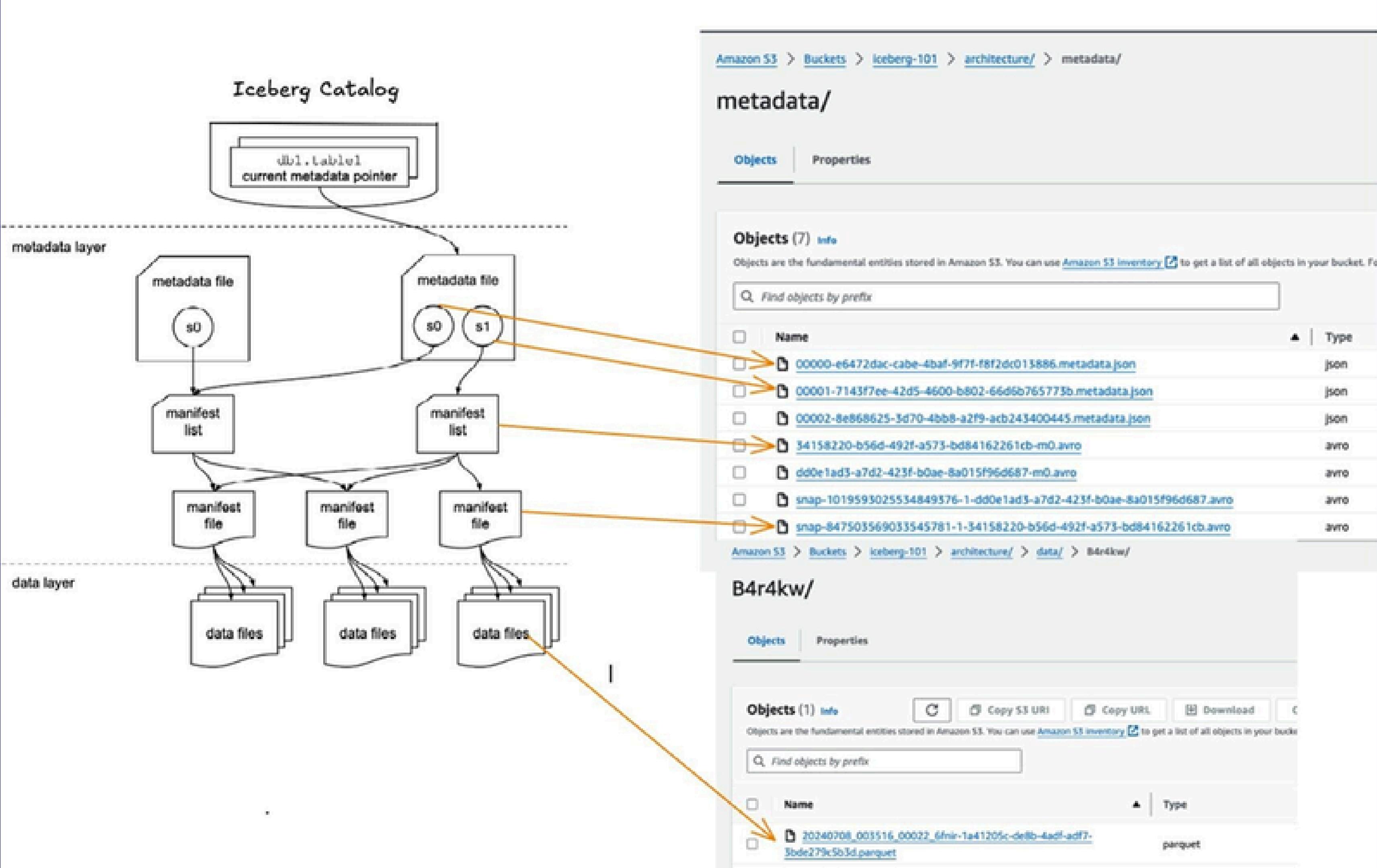
Truy vấn bất kỳ phiên bản nào phía trước của bảng

## **Schema & Partition Evolution:**

Khả năng truy vấn ngay cả khi bạn thay đổi lược đồ hay phân vùng

## **Open Source:**

Nhiều công cụ truy vấn có thể truy cập tới dữ liệu



# Storage Options

Production Database	Data Warehouse	Data Lake	Data Lakehouse
<ul style="list-style-type: none"><li>• Process small amounts of structured data</li></ul>	<ul style="list-style-type: none"><li>• Bring together large volumes of structured / semi-structured data</li><li>• Query current and historical data</li></ul>	<ul style="list-style-type: none"><li>• Process large volumes of structured, semi-structured and unstructured data</li><li>• Save on storage cost</li></ul>	<ul style="list-style-type: none"><li>• Data management &amp; discoverability features</li><li>• Low latency queries</li></ul>
<b>Use case:</b> reporting, analytics	<b>Use case:</b> reporting, analytics	<b>Use case:</b> machine learning	<b>Use case:</b> machine learning, analytics, reporting