Fundamentals of Data Engineering

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Module-1

1. Introduction to Data Engineering,
2. Data Engineering Lifecycle,
3. Data Pipeline,
4. History of Data Engineering,
5. The Data Engineer Among Other Stakeholders,
6. Business Value,
7. Translate Stakeholder Needs into Specific Requirements,
8. Data Generation in Source Systems,
9. Ingestion, Storage, Queries,
10. Modelling and Transformation,
11. Serving Data, Security,
12. Data Management, Data Architecture, Orchestration

Scenario

Every stakeholder interacts with the data ecosystem differently: Business executives want results and trends. Data scientists need structured data to build models. Developers need usable interfaces (APIs).Customers want value and privacy. You, the data engineer, ensure the whole data flow works efficiently and ethically.

Data Engineer

Wasting time & resources!

System Requirements

Functional Requirements	Nonfunctional Requirements

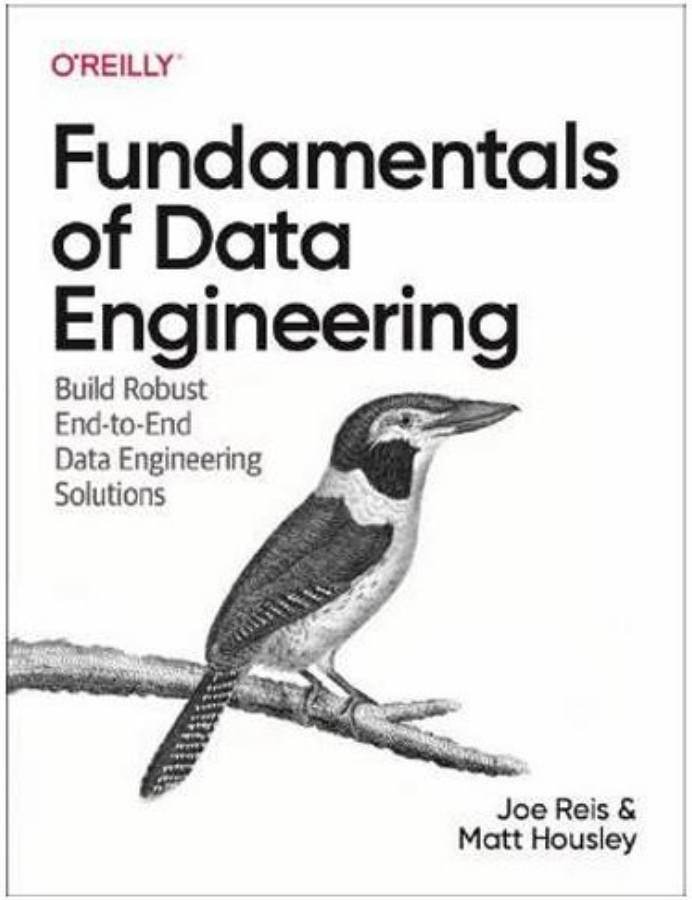
Technology Choices

- Ingestion tool
- Storage system
- Transformation tool

Data engineering

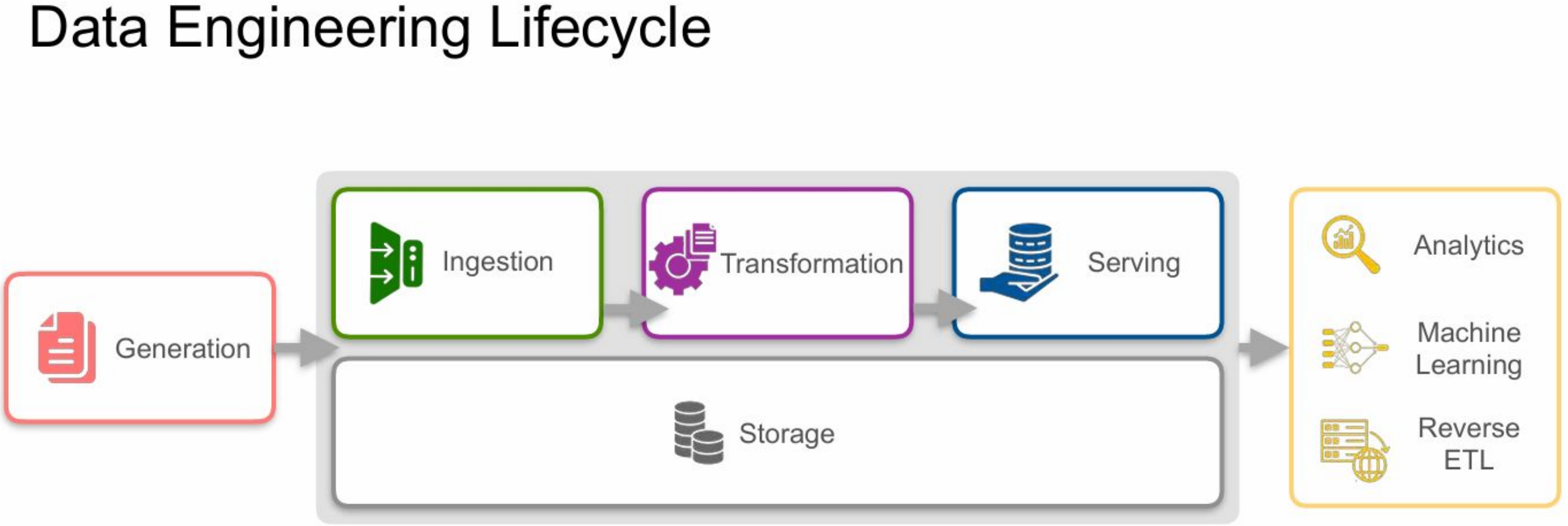
RV UNIVERSITY
Go, change the world

Data Engineering

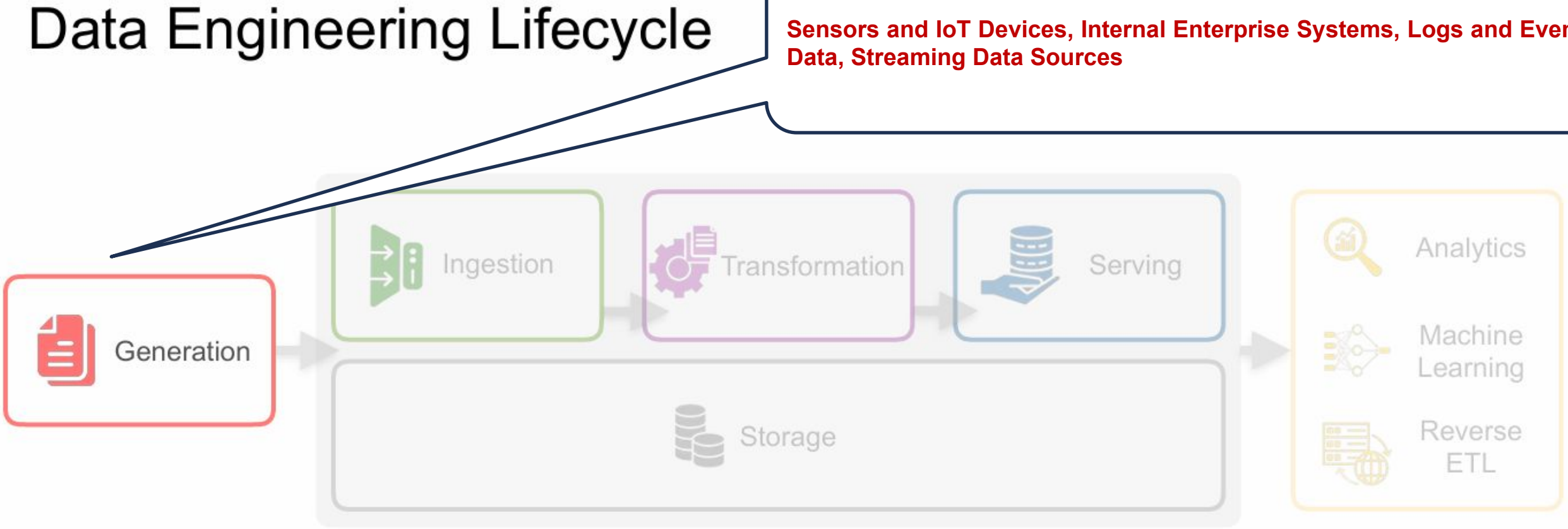


"Data engineering is the development, implementation, and maintenance of systems and processes that take in raw data and produce high-quality, consistent information that supports downstream use cases, such as analysis and machine learning. Data engineering is the intersection of security, data management, DataOps, data architecture, orchestration, and software engineering."

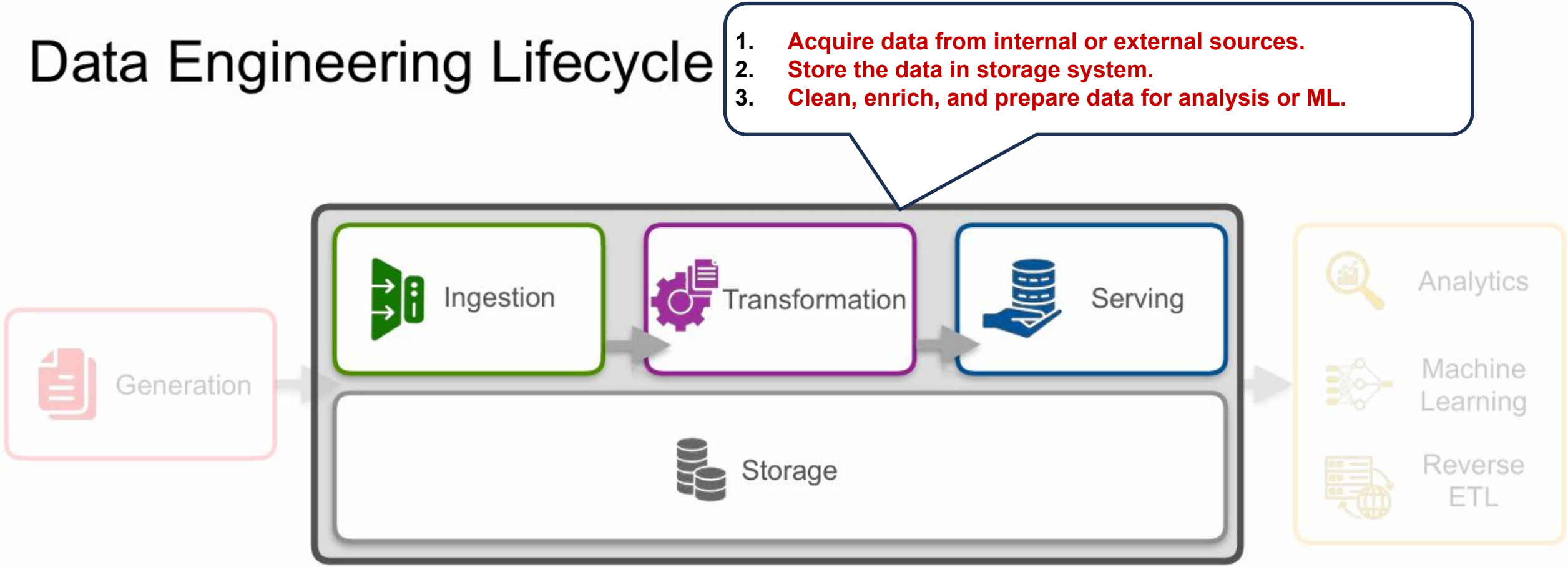
Data Engineering Lifecycle



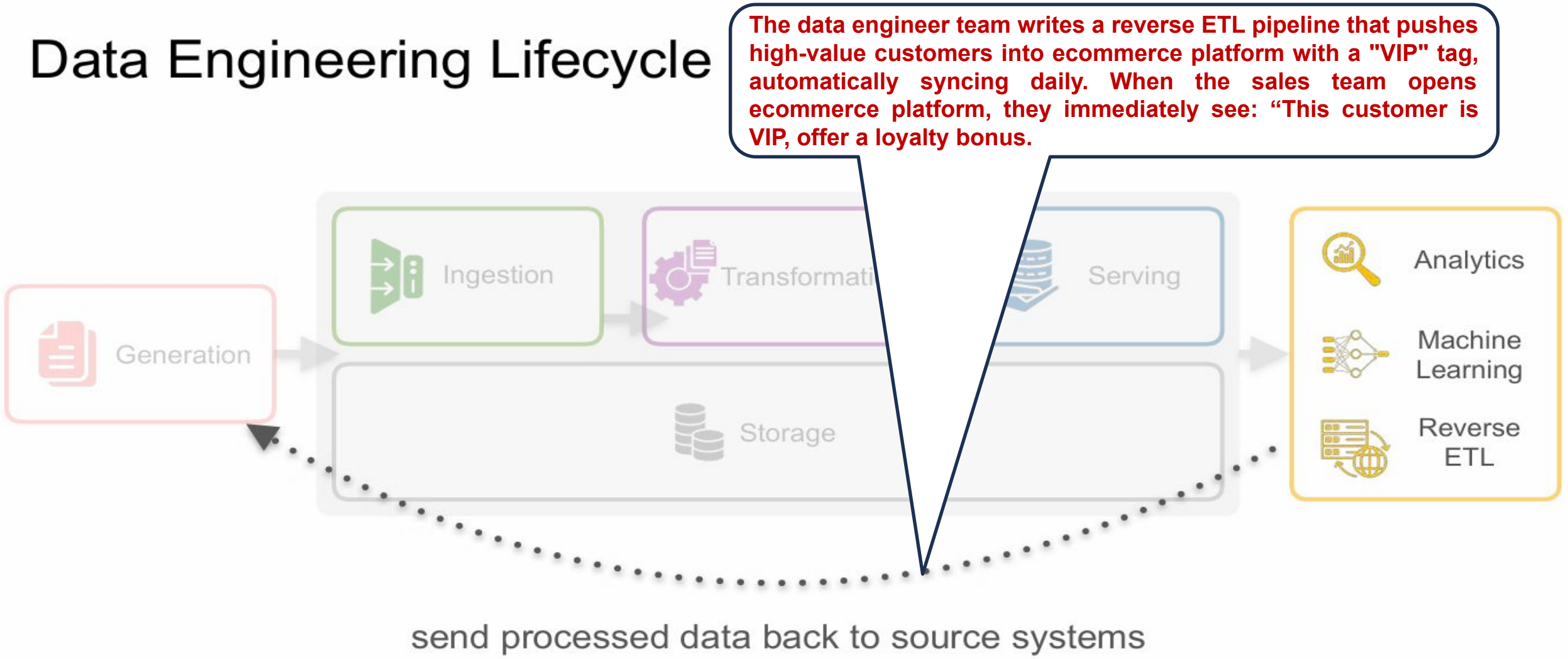
Data Engineering Lifecycle



Data Engineering Lifecycle

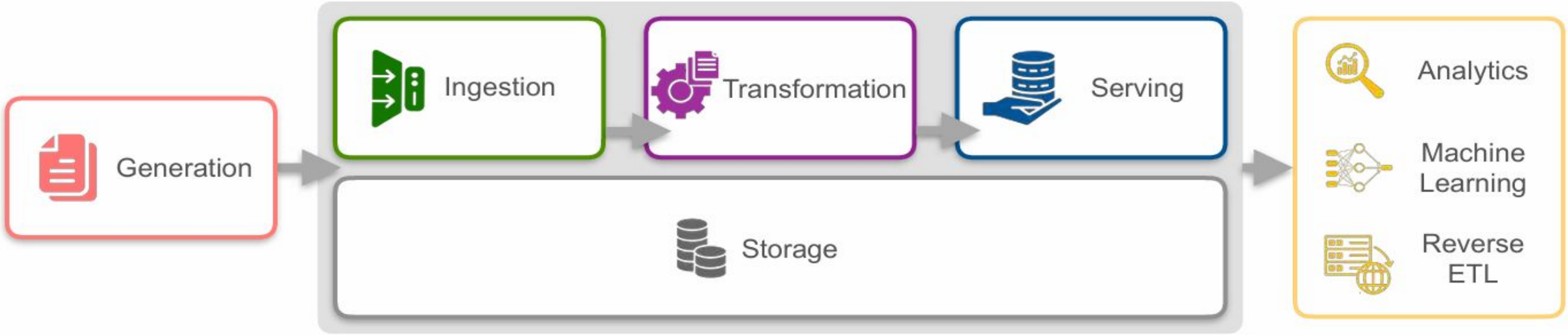


Data Engineering Lifecycle



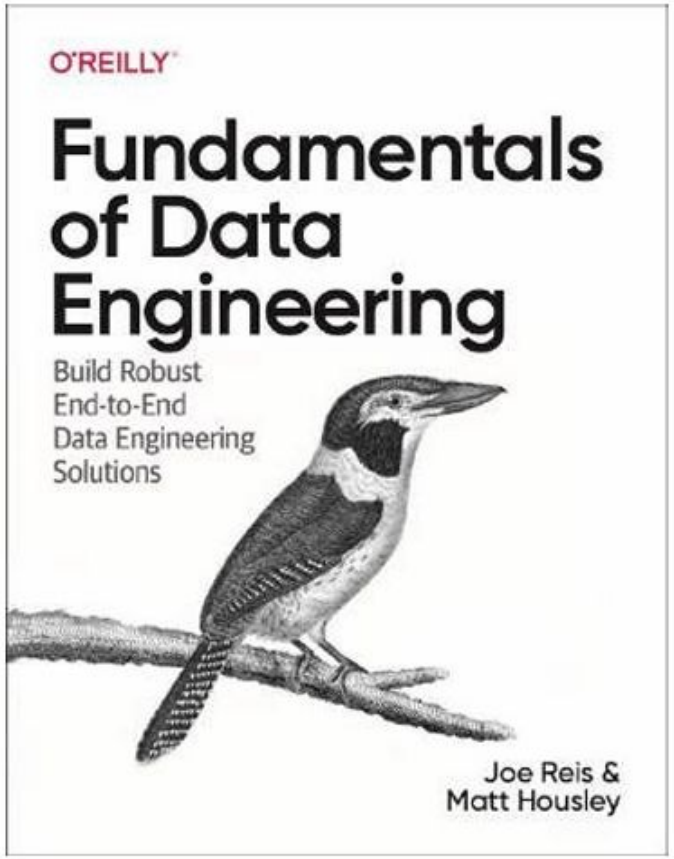
Data Pipeline

Data Pipeline



Data Pipeline The combination of architecture, systems, and processes that move data through the stages of the data engineering lifecycle.

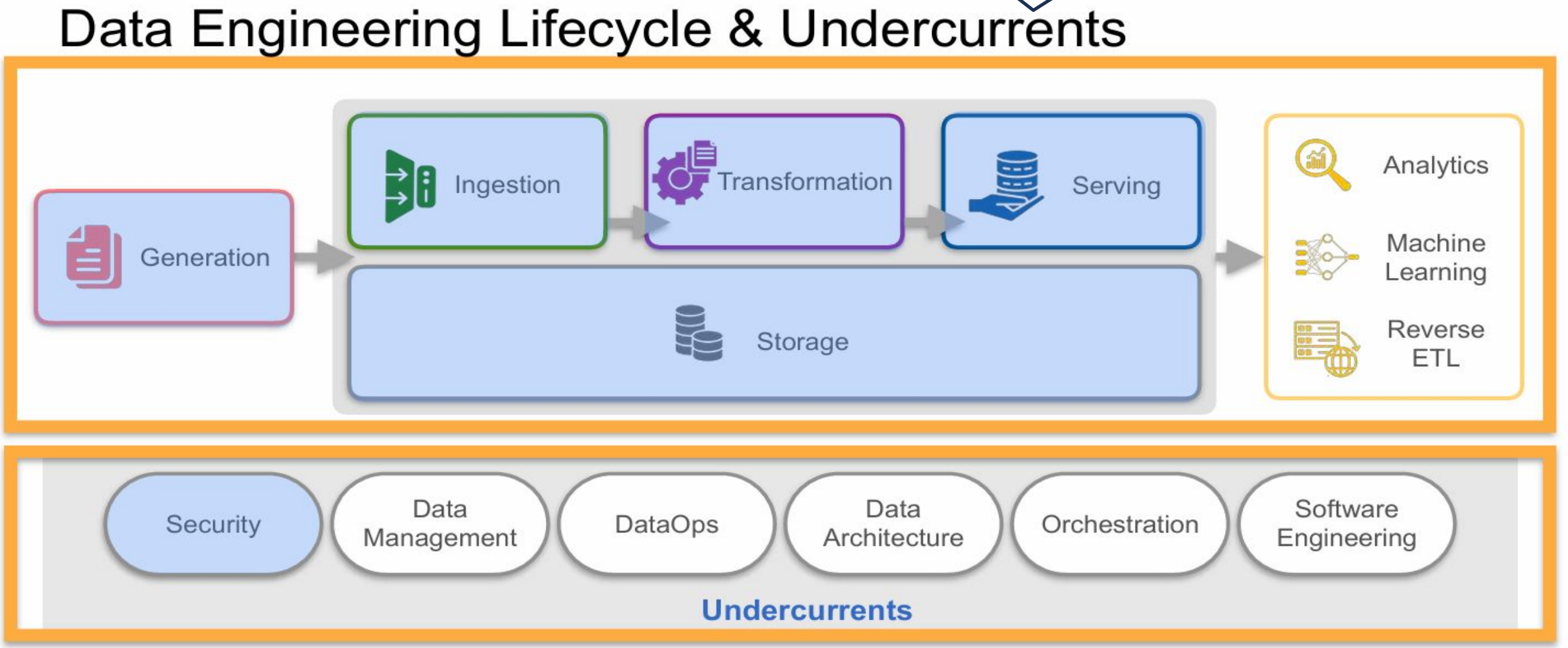
Data Engineering



"Data engineering is the development, implementation, and maintenance of systems and processes that take in raw data and produce high-quality, consistent information that supports downstream use cases, such as analysis and machine learning. Data engineering is the intersection of Security, Data Management, DataOps, Data Architecture, Orchestration, and Software Engineering."

Data Engineering Lifecycle and Undercurrents

Undercurrents: Hidden or less-visible forces that significantly influence data engineering practices and outcomes.



RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

A Brief History of Data Engineering

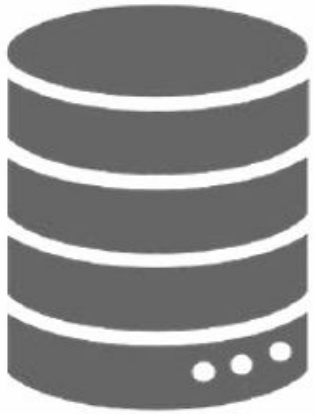
History of Data Engineering

History of Data Engineering

1960s

Computers

Computerized Database



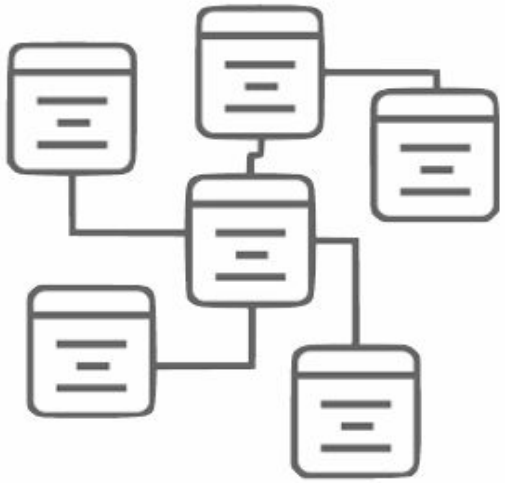
History of Data Engineering

1960s

1970s

IBM

Relational Databases



Structured Query Language

History of Data Engineering

1960s

1970s

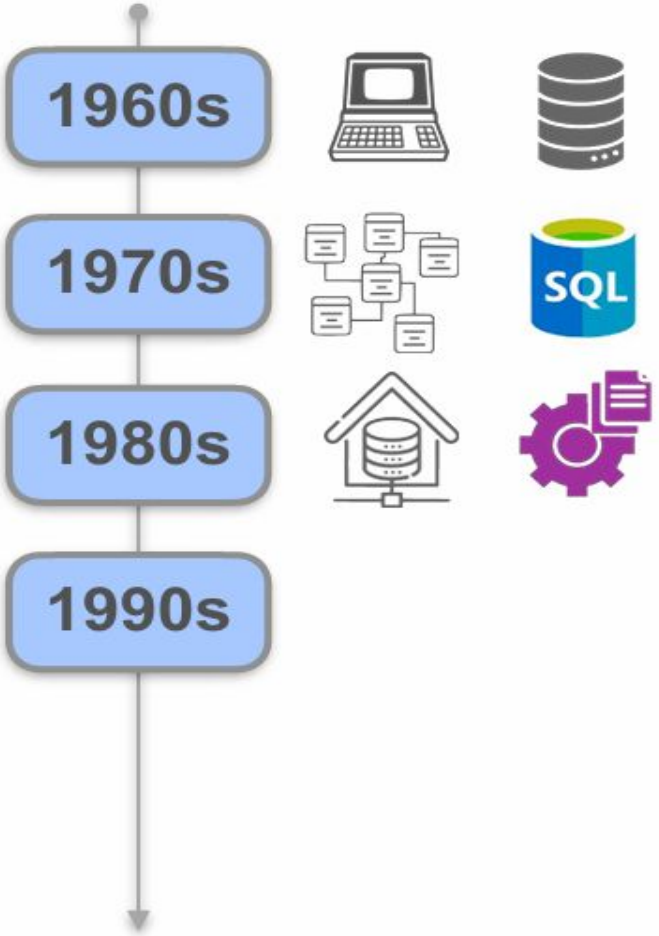
1980s

Bill Inmon

Data Warehouse

Transforming Data

History of Data Engineering

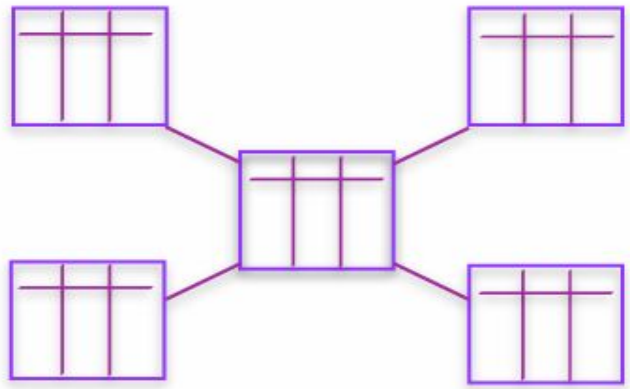


Business Intelligence

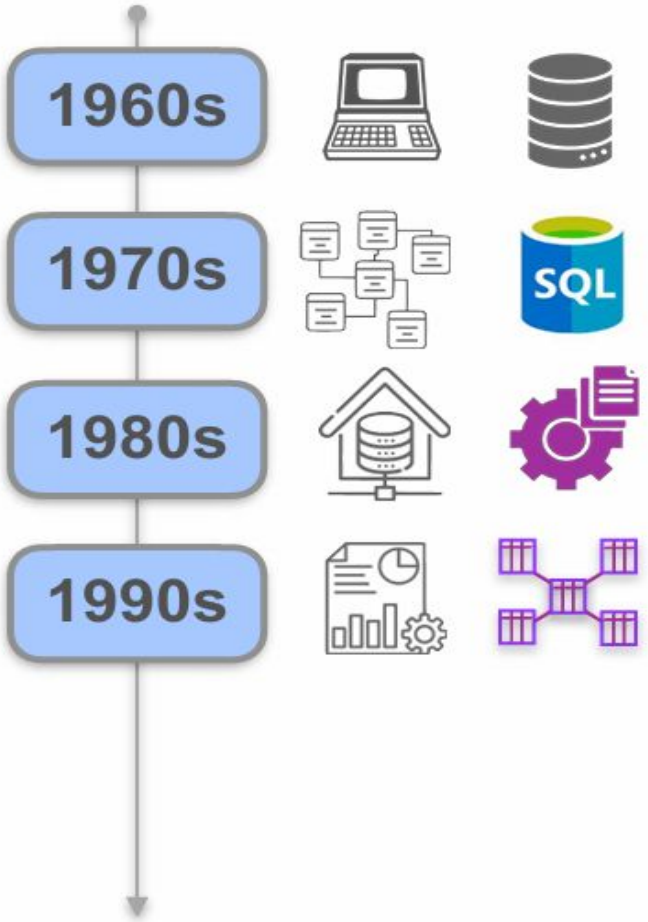
Bill Inmon

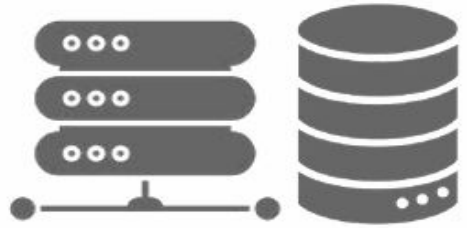
Ralph Kimball

Data Modeling

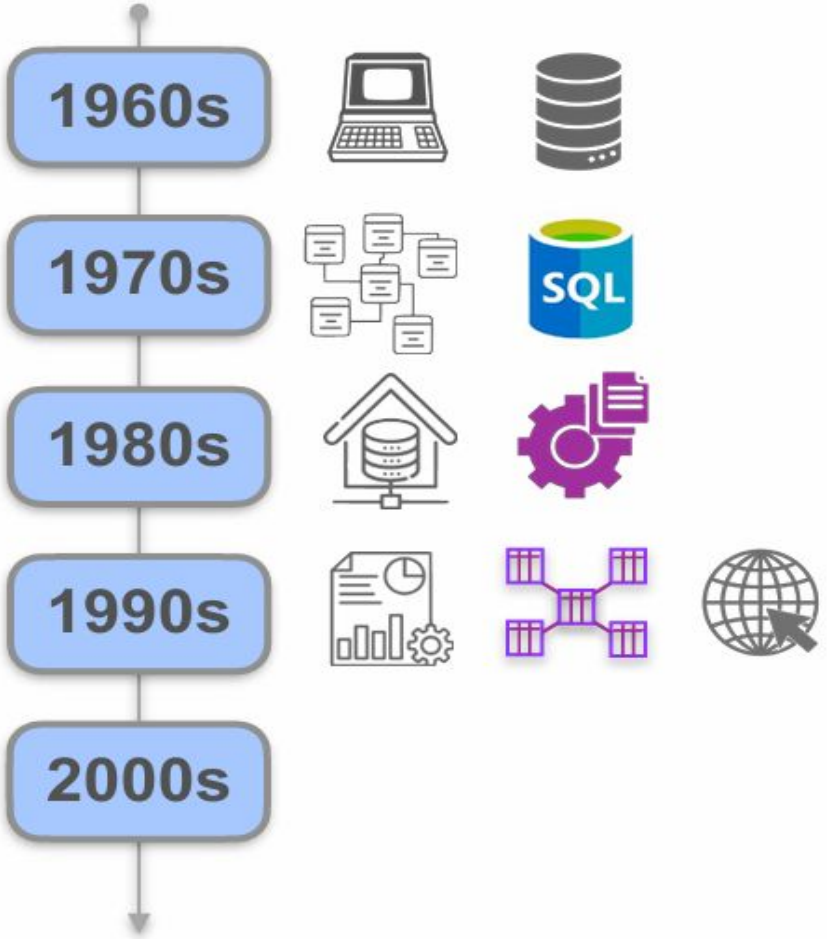


History of Data Engineering





History of Data Engineering



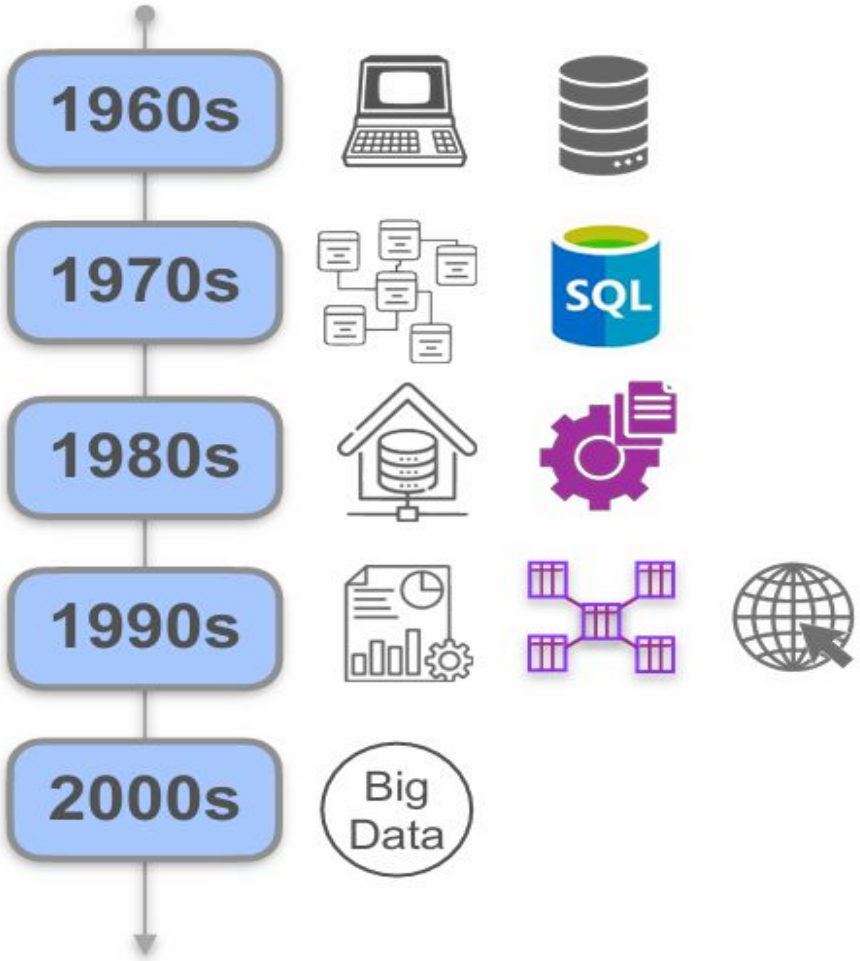
yahoo!

Google

amazon

Big Data

History of Data Engineering

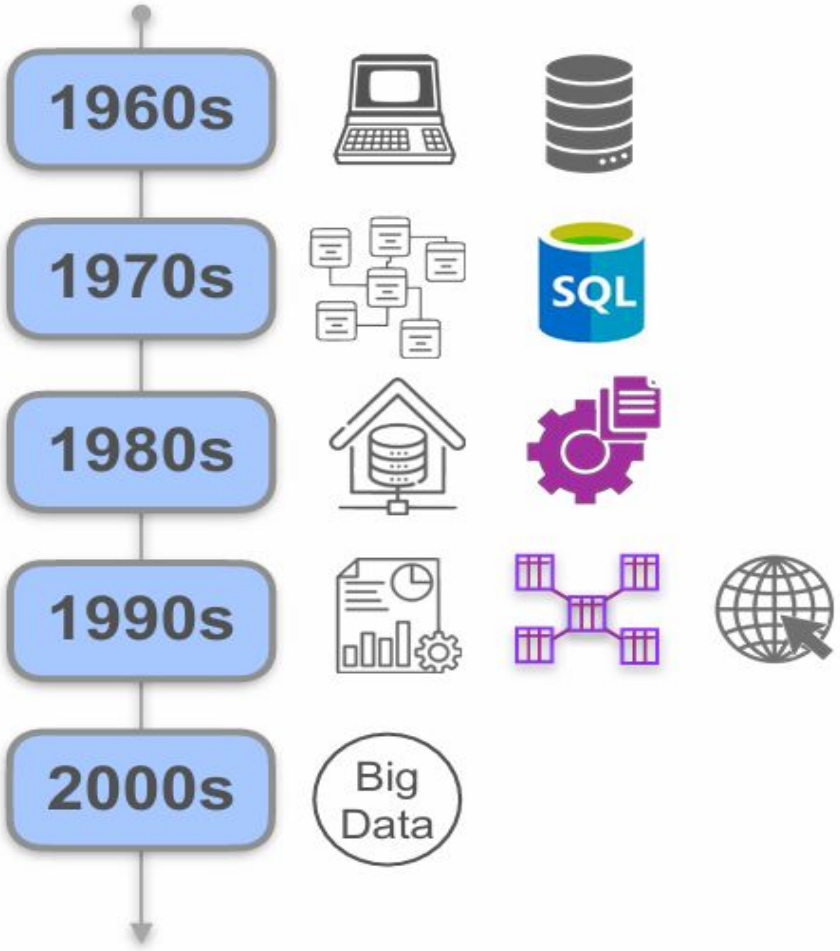


The "Big Data" Era

"extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions."

Velocity Variety Volume

History of Data Engineering



2004

Google

MapReduce: Simplified Data Processing on Large Clusters

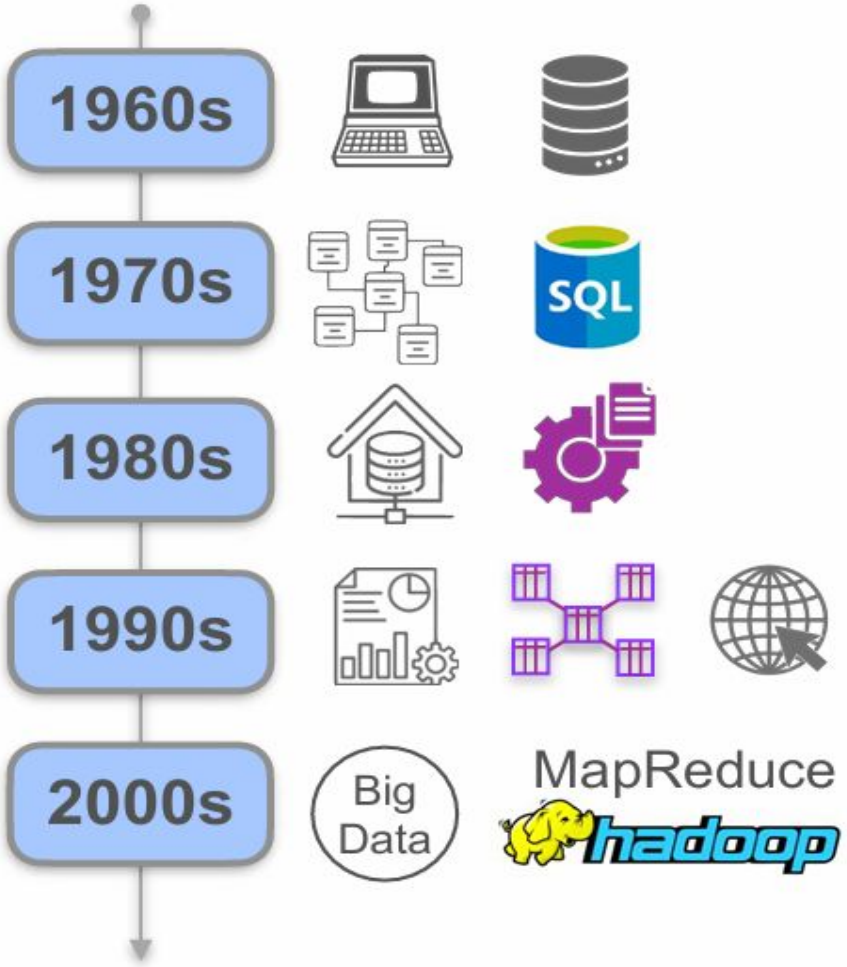
2006

yahoo!

hadoop

The "Big Data Engineer" Era

History of Data Engineering



Pay-as-you-go resource marketplace

Amazon EC2

Amazon S3

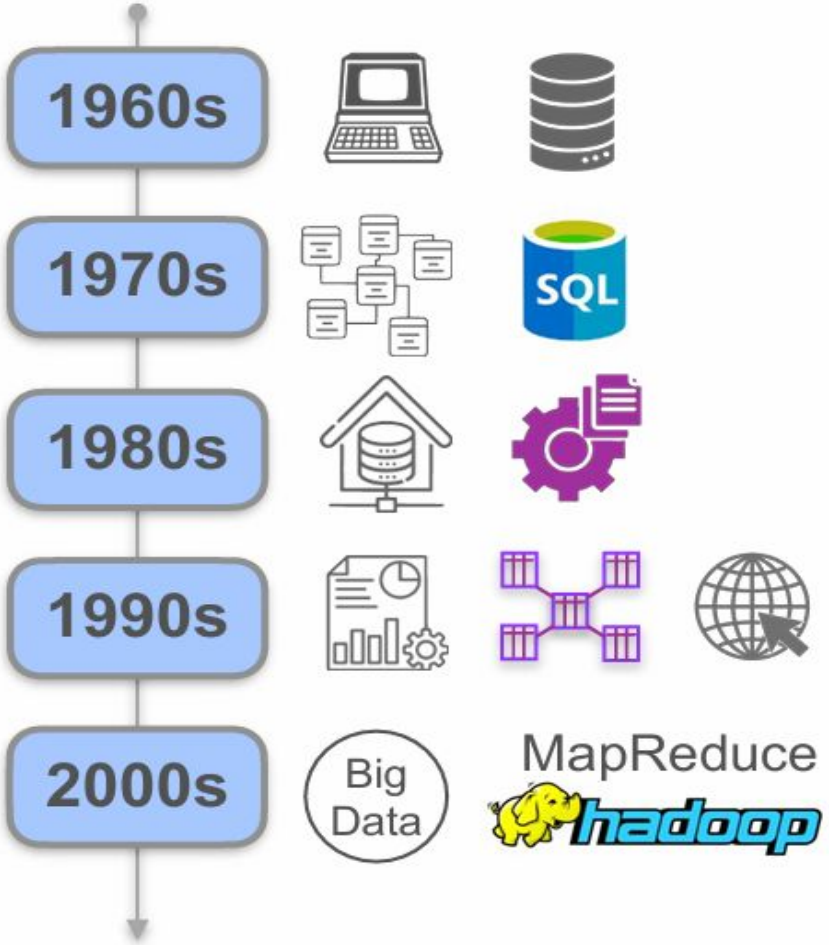
Amazon DynamoDB

aws

Amazon Web Services

The first popular public cloud

History of Data Engineering



Public Cloud

aws

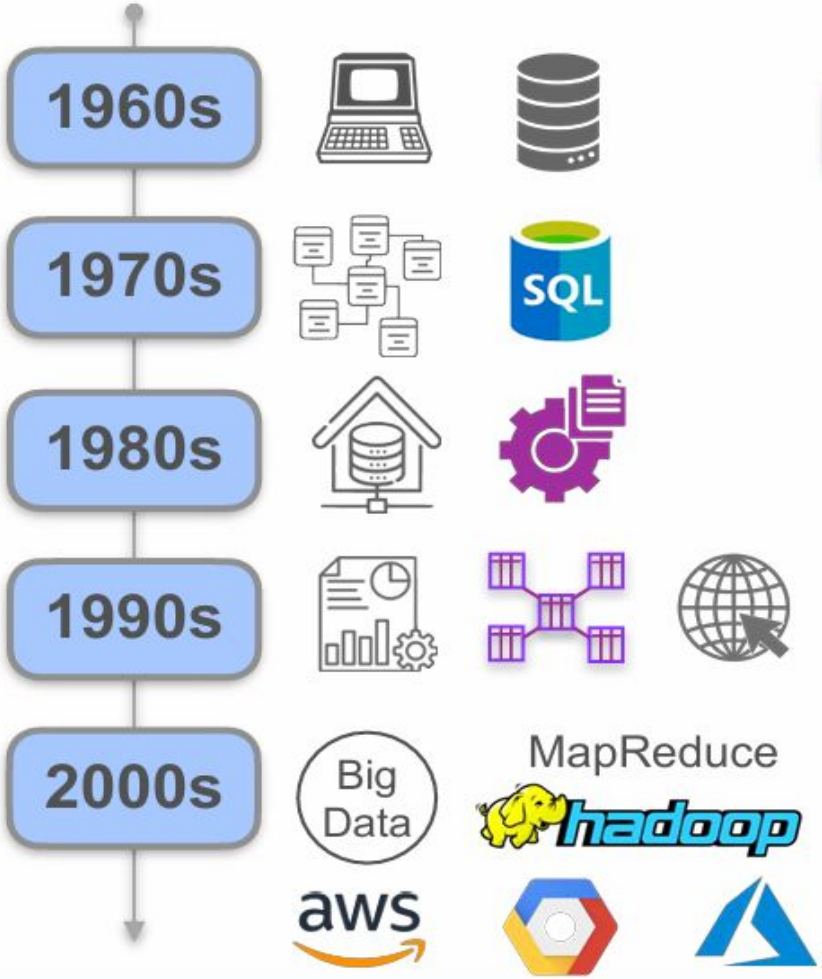
Google Cloud Platform

Azure

Public Cloud & Early big data tools :

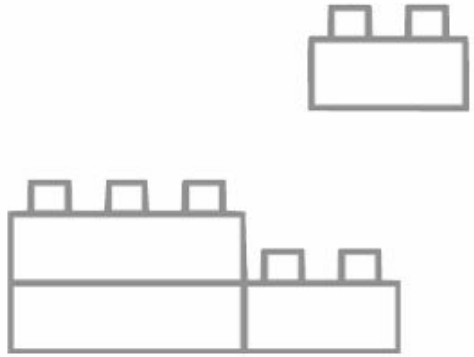
Foundation for today's data ecosystem

History of Data Engineering



2010s - present

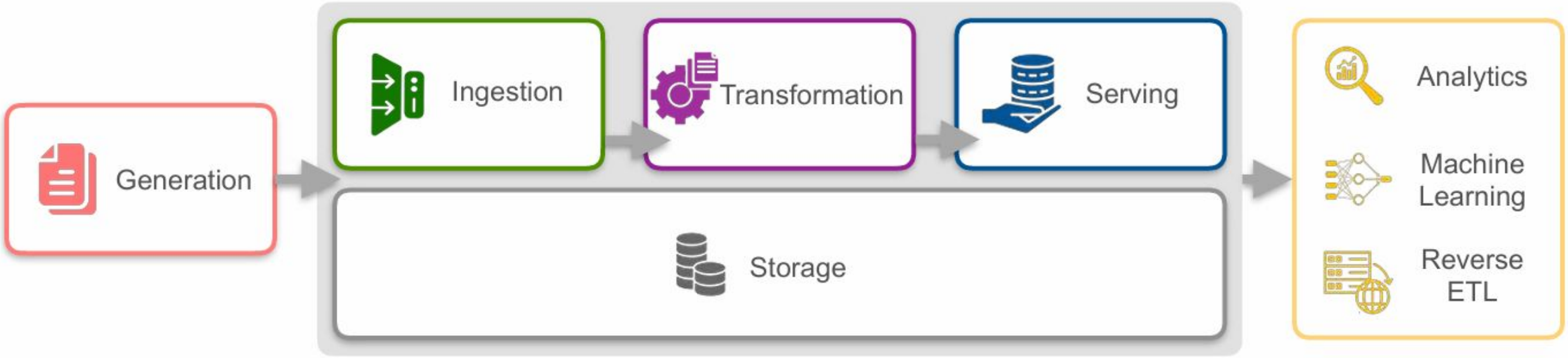
Abstract Simplified Tools

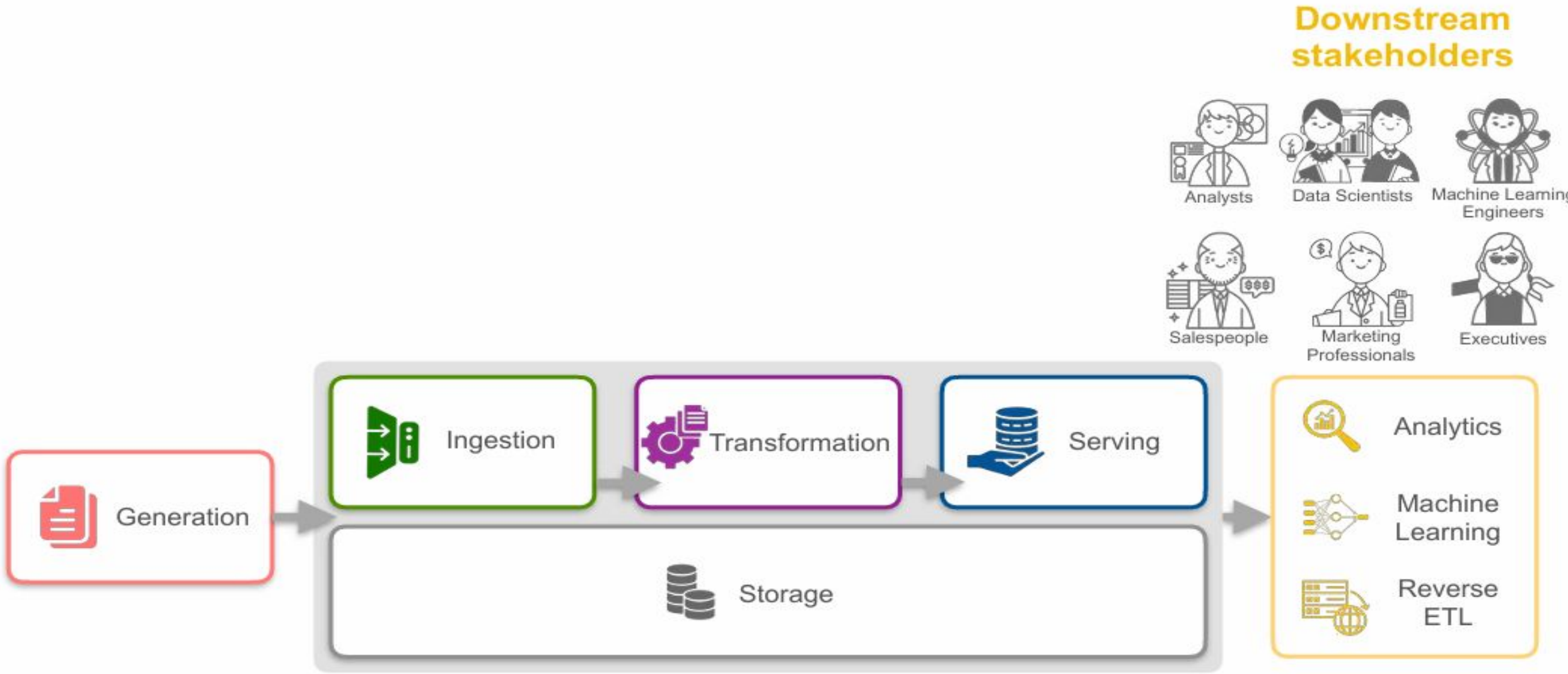


RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

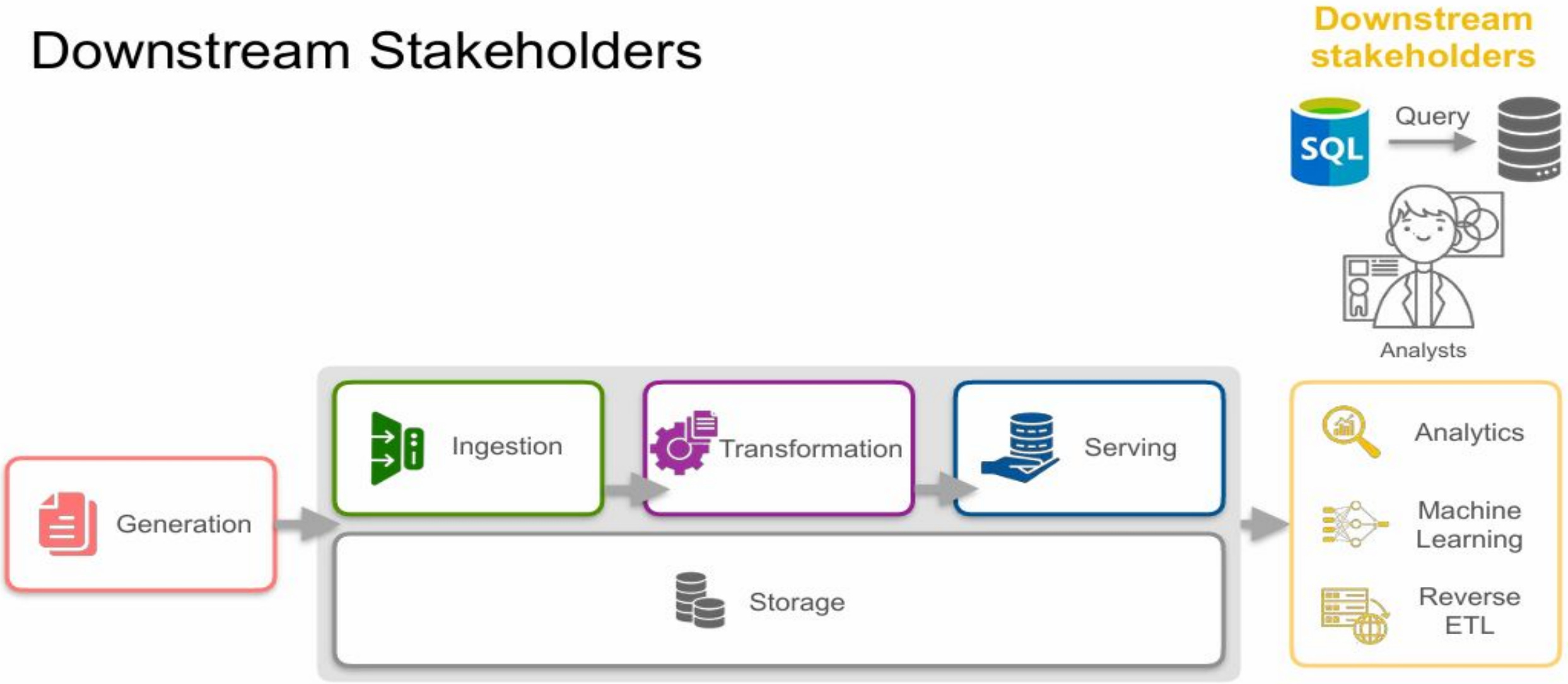
The Data Engineer Among Other Stakeholders

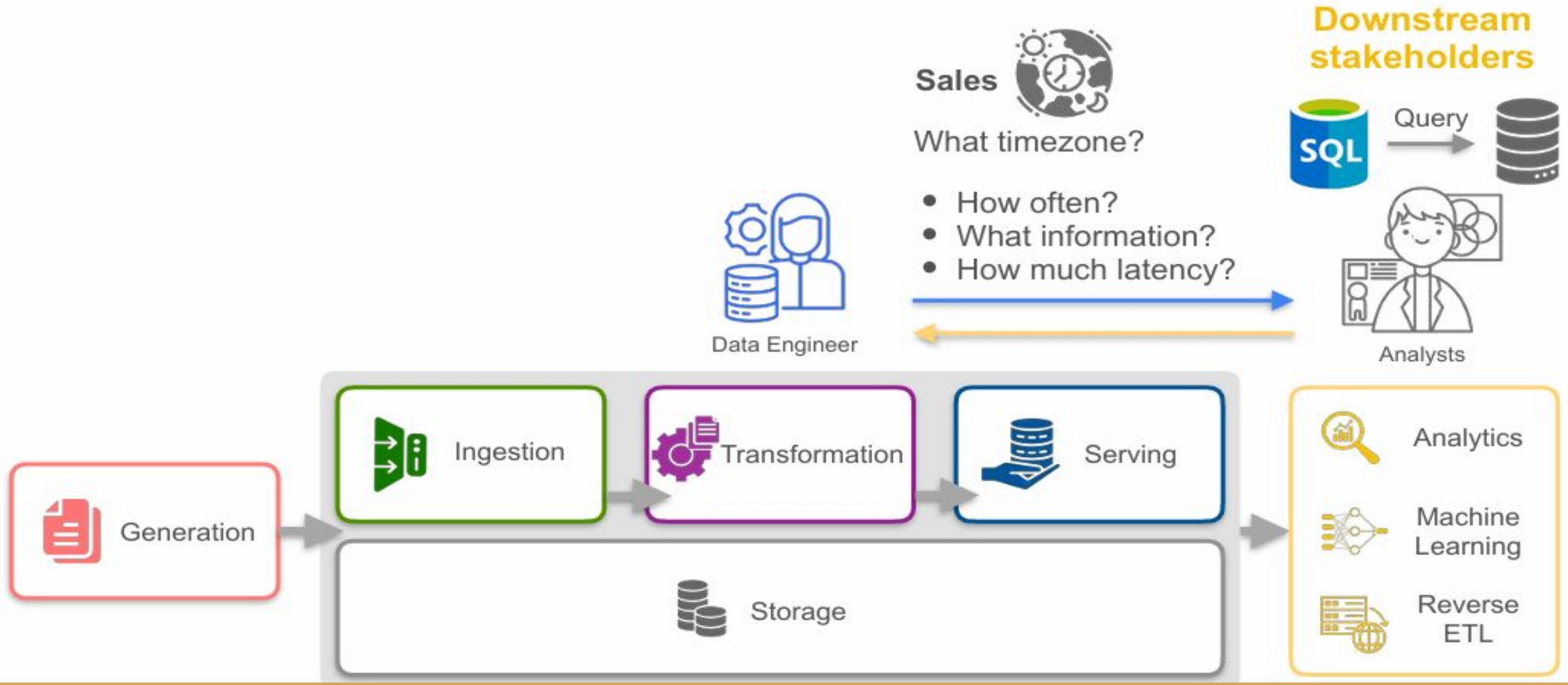
Downstream Use Cases

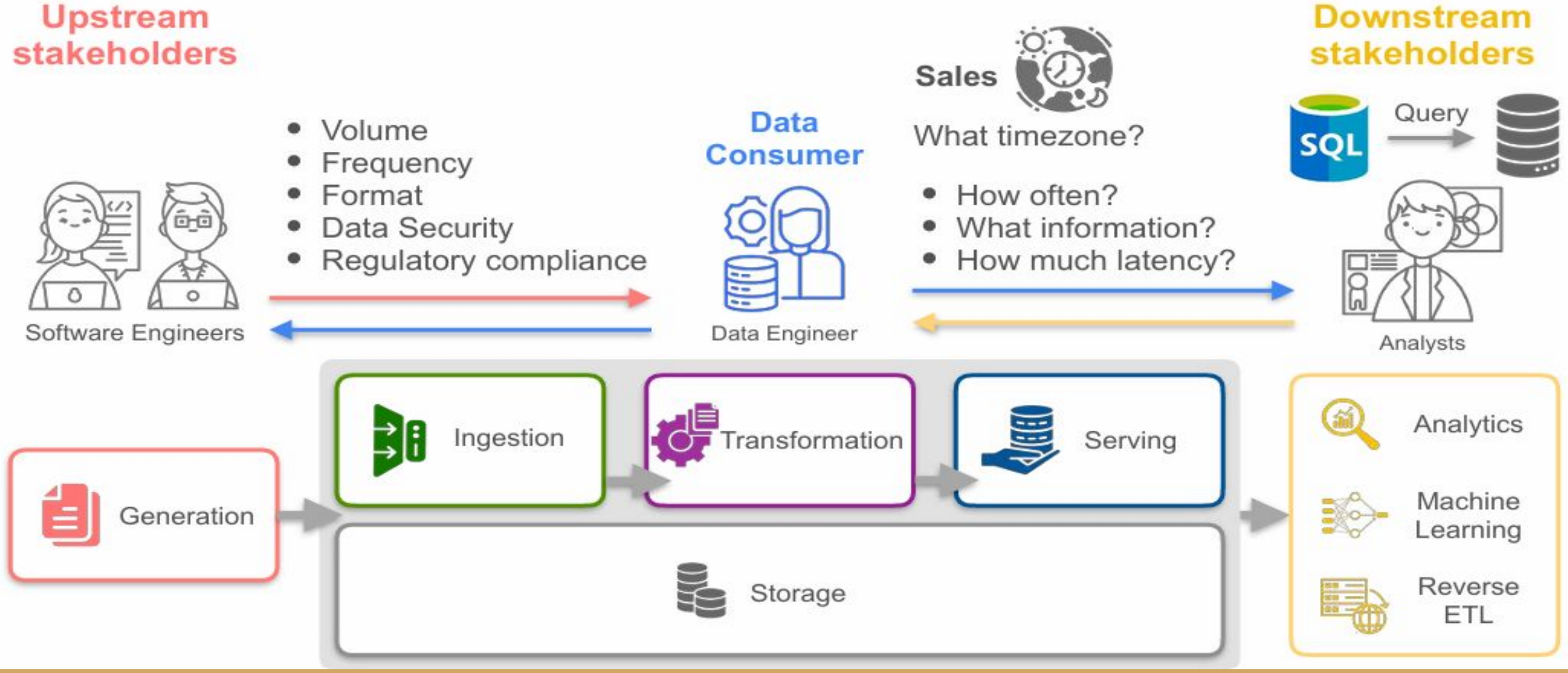


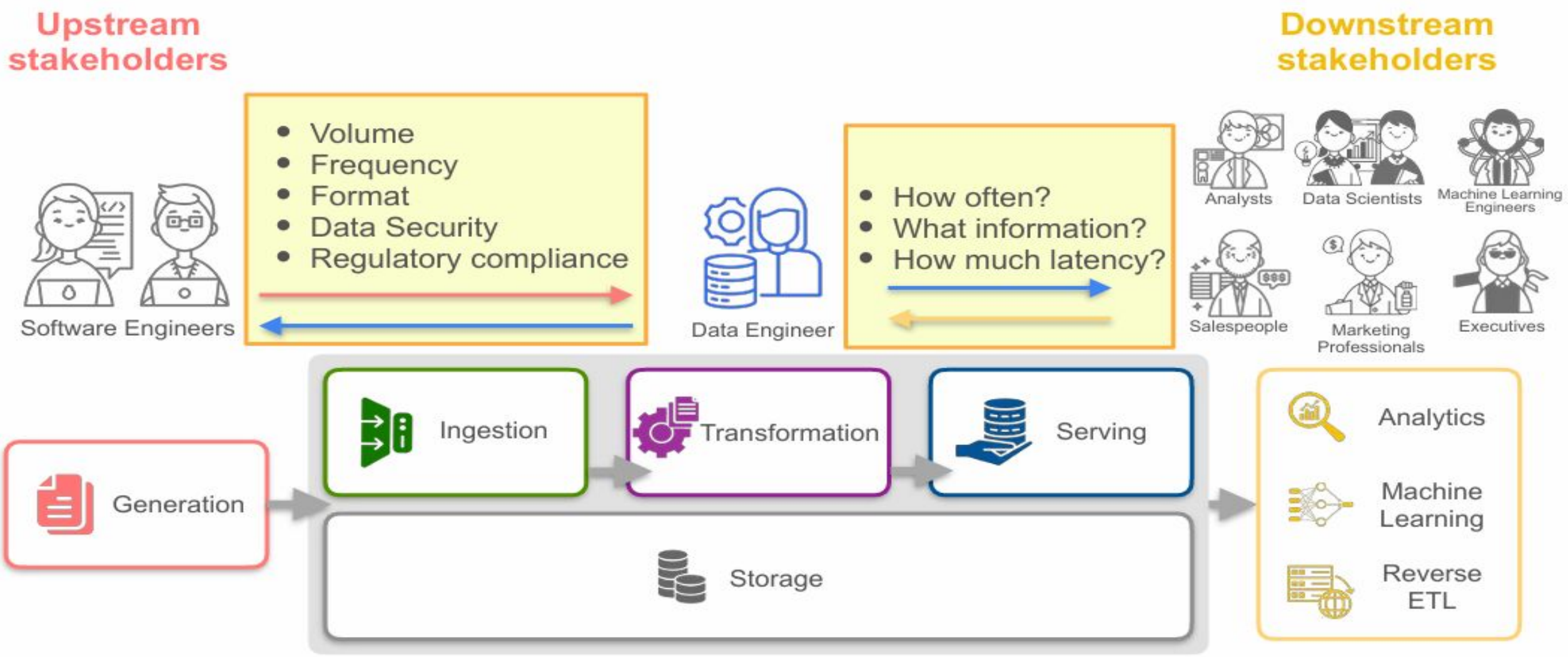

Downstream stakeholders
Analysts
Data Scientists
Machine Learning Engineers
Salespeople
Marketing Professionals
Executives
Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine Learning
Reverse ETL

Downstream Stakeholders




Downstream stakeholders
Sales
What timezone?
SQL
Query
How often?
What information?
How much latency?
Data Engineer
Analysts
Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine Learning
Reverse ETL


Upstream stakeholders
Volume
Frequency
Format
Data Security
Regulatory compliance
Software Engineers
Data Consumer
Data Engineer
Sales
What timezone?
How often?
What information?
How much latency?
Downstream stakeholders
SQL
Query
Analysts
Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine Learning
Reverse ETL


Upstream stakeholders
Downstream stakeholders
Software Engineers
Volume
Frequency
Format
Data Security
Regulatory compliance
Data Engineer
How often?
What information?
How much latency?
Analysts
Data Scientists
Machine Learning Engineers
Salespeople
Marketing Professionals
Executives
Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine Learning
Reverse ETL

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Business Value

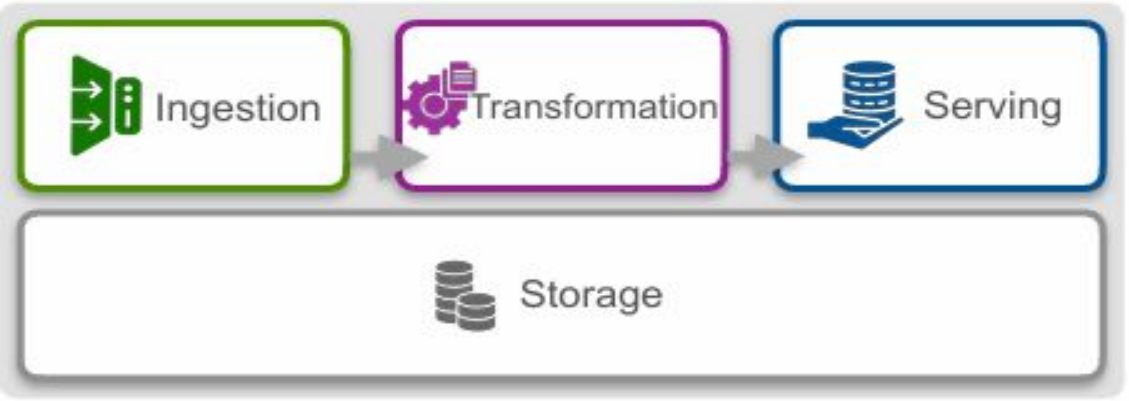
Business Value

Goal: Revenue Growth

Value Created!

Business Value

Goal: Revenue Growth



No Value!

Business Value

Is your work helping them achieve their goals?

Multiple forms for business value


Analysts


Machine Learning Engineers


Marketing Professionals

Business Value

Multiple forms for business value

- Increased Revenue
- Cost Savings
- Improved efficiency
- Launch a product

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

System Requirements

Requirements

Business Requirements

High level goals of the business

For example: grow revenue, increase user base

Stakeholder Requirements

Needs of individuals within the organization

Things they need to get their job done well

System Requirements

- **Functional Requirements** — The "WHAT"
- **Non-Functional Requirements** — The "HOW"

Requirements

Functional Requirements

What the system needs to be able to do

- Provide regular updates to a database
- Alert a user about an anomaly in the data

Non-Functional Requirements

How the system accomplishes what it needs to do

- Technical specifications of an ingestion or orchestration or storage approach
- How you'll meet the end user's needs

Requirements Gathering

Business & Stakeholder Requirements

Features & Attributes

Memory & Storage Capacity

Cost & Security Constraints

Gather your system requirements

Translate

Data Engineer

High-level goals & needs

Analysts

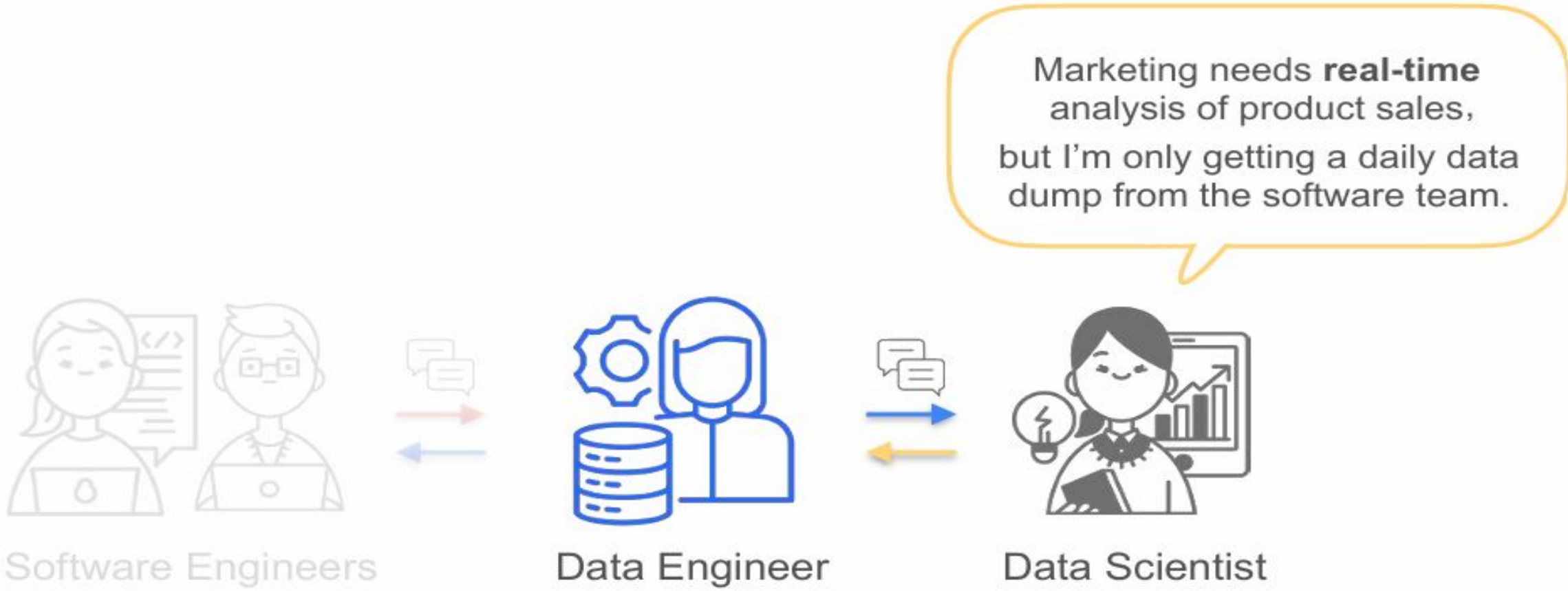
ML Engineers

Marketing Professionals

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Translate Stakeholder Needs into Specific Requirements

Conversation with Data Scientist



Conversation with Data Scientist

- Build in automatic checks on the ingested data
- Know about changes or disruptions before they happen

Problems with schema changes & other anomalies in the data

Software Engineers

Data Engineer

Data Scientist

Conversation with Source Owners

Conversation with Data Scientist

Functional Requirement

Ingest, transform, & serve data in the format required

Non-Functional Requirement

Make data available some time after it is recorded

Lots of data cleaning & processing

Conversation with Data Scientist

What is "real-time"?

- monthly basis
- daily, hourly, minutes, seconds, ...

The marketing team needs the data in real-time

Software Engineers

Data Engineer

Data Scientist

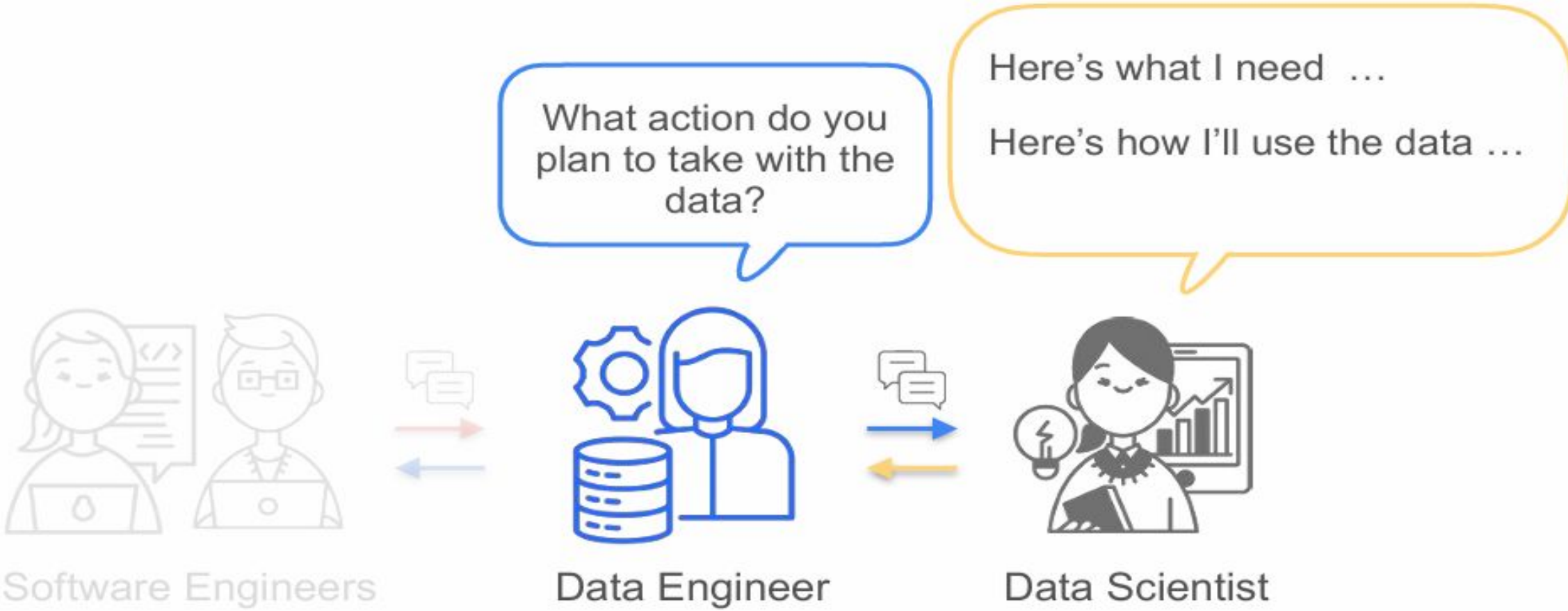
Conversation with Data Scientist

Key Tactic:

Ask stakeholder what action they plan to take with the data

Not the same as asking what they need!

Conversation with Data Scientist

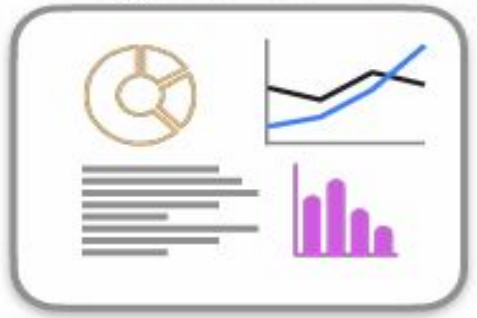


Conversation with Data Scientist

A recommender system

Provide product recommendations in near real-time

An analytics dashboard



?

Software Engineers

Data Engineer

Data Scientist

Conversation with Data Scientist

1. Learned about existing solutions and pain points
2. Started to identify some of your system requirements
3. Identified stakeholders to talk to:
   - Marketing team
   - Software engineering team

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Thinking Like a Data Engineer

Thinking Like a Data Engineer

1 2 3 4

Identify business goals & stakeholder needs

Define system requirements

Choose tools & technologies

Build, evaluate, iterate & evolve

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

Business Goals

Stakeholders' Needs

Thinking Like a Data Engineer

1 Identify business goals & stakeholder needs

2 **Define system requirements**

3 Choose tools & technologies

4 Build, evaluate, iterate & evolve

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

Confirm with Stakeholders

Thinking Like a Data Engineer

1

Identify business goals & stakeholder needs

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

2

Define system requirements

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

3

Choose tools & technologies

1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

4

Build, evaluate, iterate & evolve

Cost-Benefit Analysis

Thinking Like a Data Engineer

Identify business goals & stakeholder needs	Define system requirements	Choose tools & technologies	**Build, evaluate, iterate & evolve**
1. Identify business goals & stakeholders you will serve 2. Explore existing systems and stakeholder needs 3. Ask stakeholders what actions they will take with the data product	1. Translate stakeholder needs to functional requirements 2. Define non-functional requirements 3. Document and confirm requirements with stakeholders	1. Identify tools & tech to meet non-functional requirements 2. Perform cost / benefit analysis and choose between comparable tools & tech 3. Prototype and test your system, align with stakeholder needs	1. Build & deploy your production data system 2. Monitor, evaluate, and iterate on your system to improve it 3. Evolve your system based on stakeholder needs

Thinking Like a Data Engineer

Identify business goals & stakeholder needs

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

Define system requirements

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

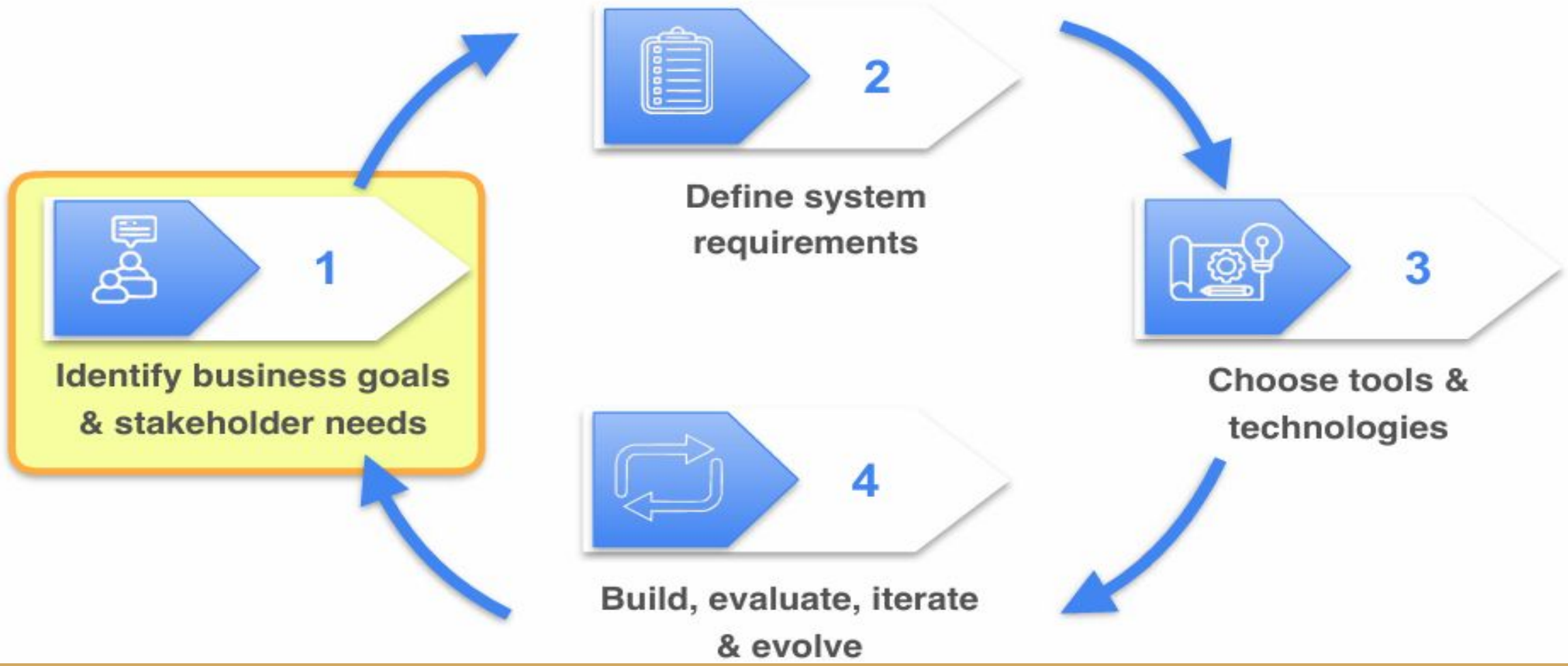
Choose tools & technologies

1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

Build, evaluate, iterate & evolve

1. Build & deploy your production data system
2. Monitor, evaluate, and iterate on your system to improve it
3. Evolve your system based on stakeholder needs

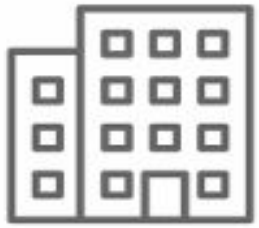
Thinking Like a Data Engineer



RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Data Engineering on the Cloud

Location



On-Premises

Migrating

Cloud

- Regulatory concerns
- Legacy systems

Hybrid

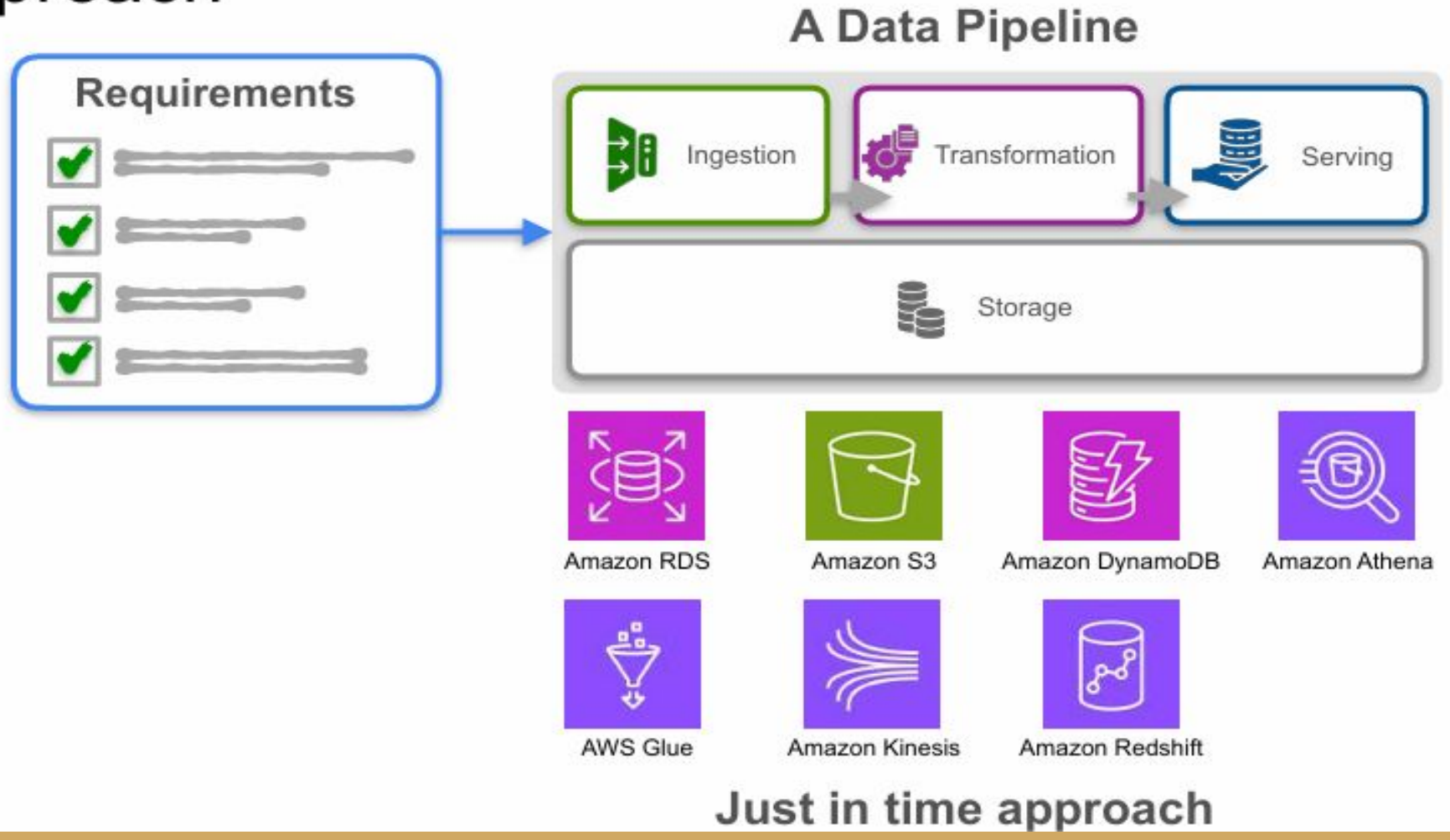
Public Cloud

Google Cloud Platform

Microsoft Azure

Specialization Approach

Cloud-first approach

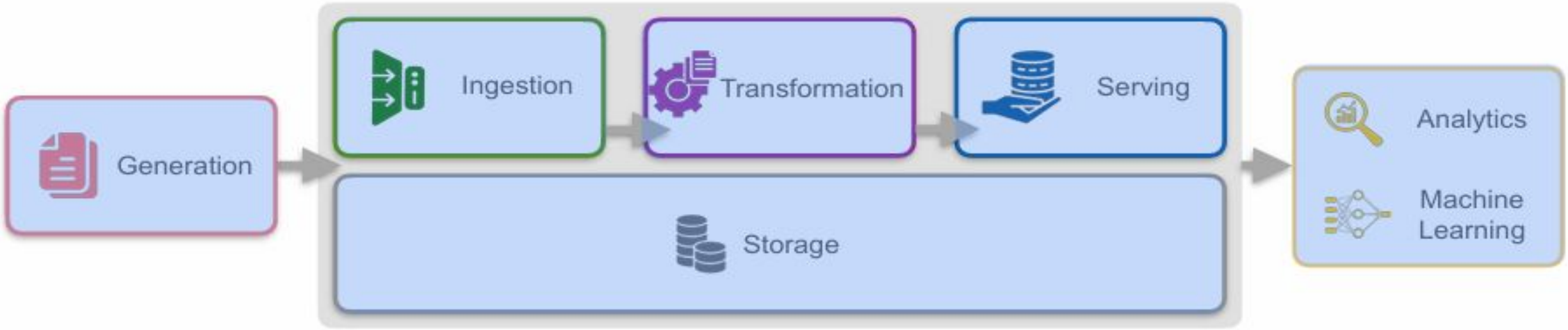


Just in time approach

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

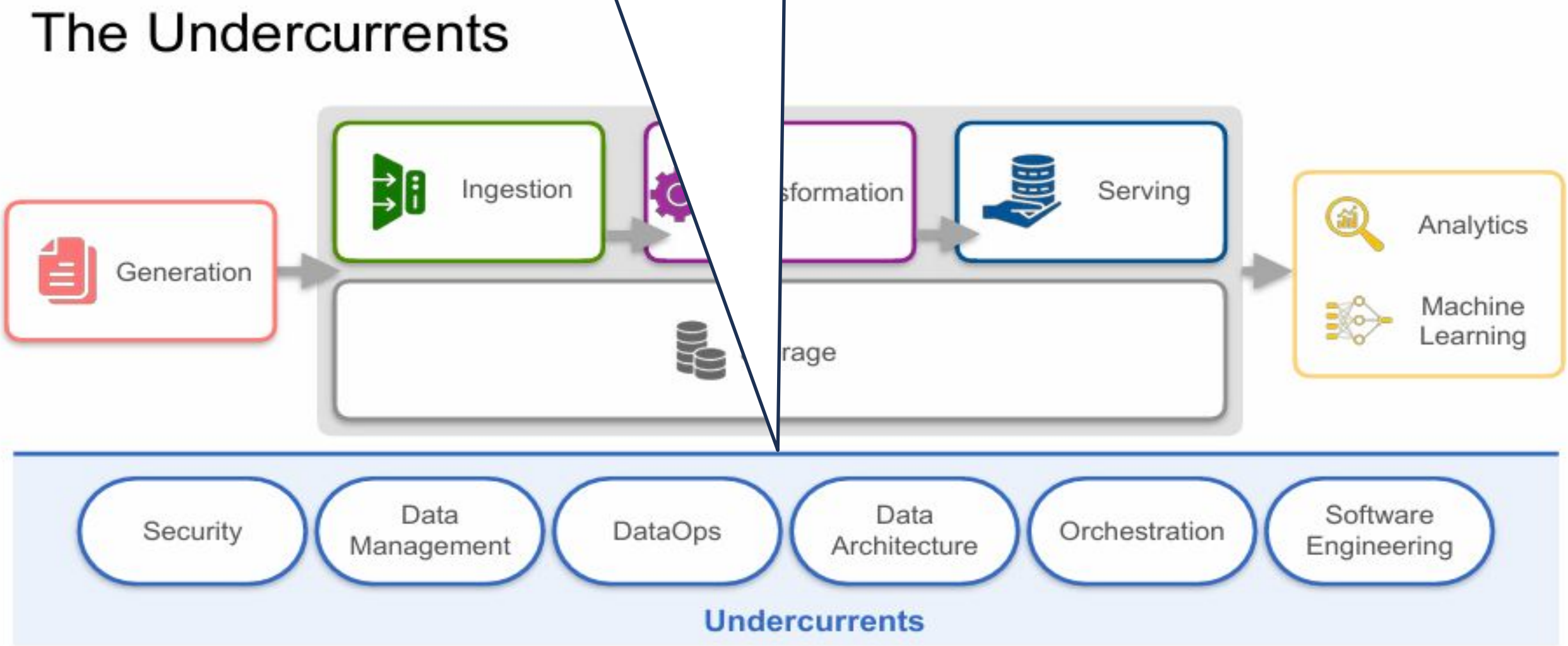
The Data Engineering Lifecycle & Undercurrents

The Data Engineering Lifecycle



Get raw data → Turn it into something useful → Make it available for downstream use cases

Undercurrents: Hidden or less-visible forces that significantly influence data engineering practices and outcomes.

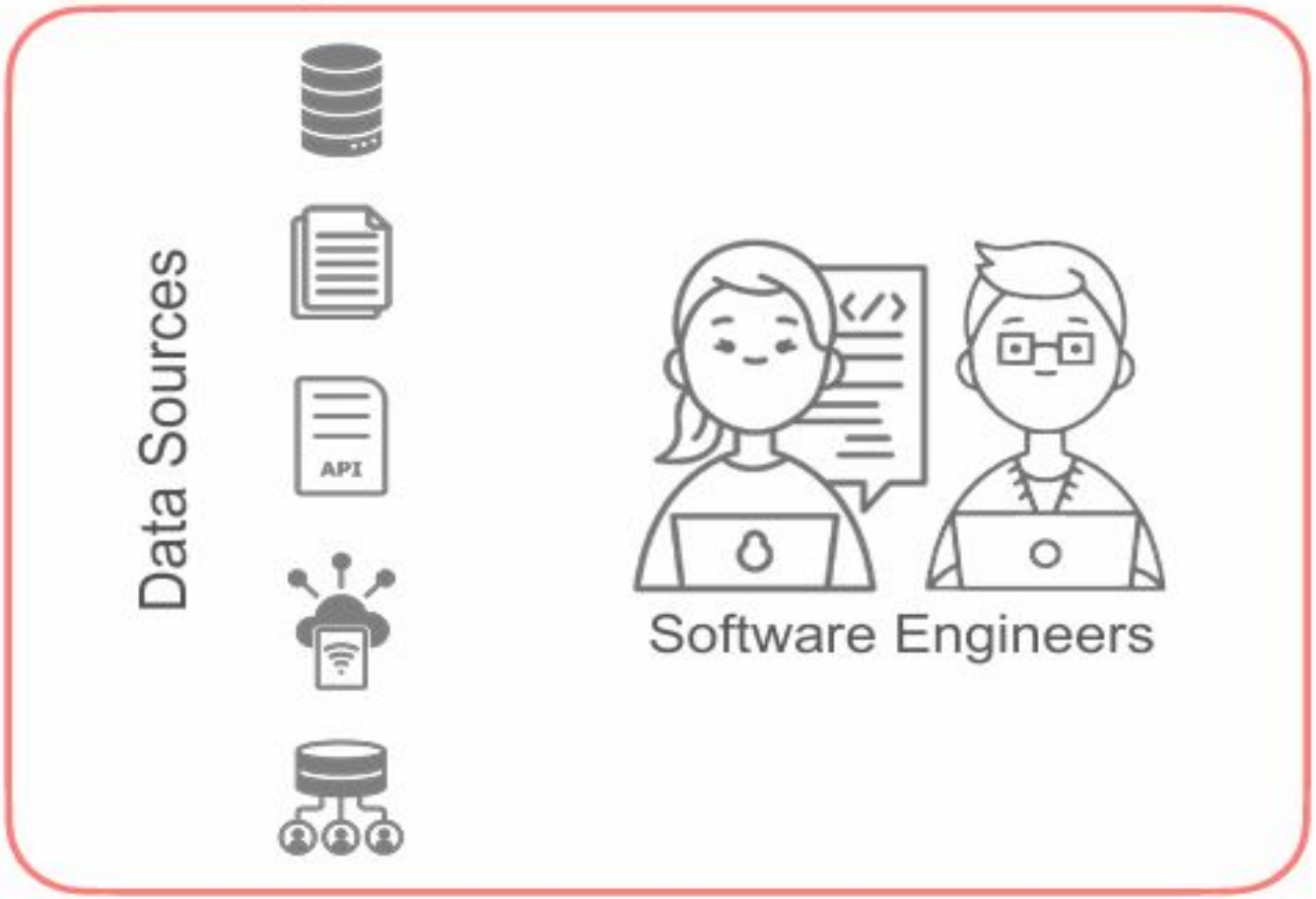


RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Data Generation in Source Systems

In an e-commerce company, data is generated from a wide range of source systems, each contributing to the overall digital ecosystem. These sources are broadly categorized into Transactional Systems, Marketing Platforms, and External Data Sources. All this data is maintained by internal department ,software engineer and third party platform.

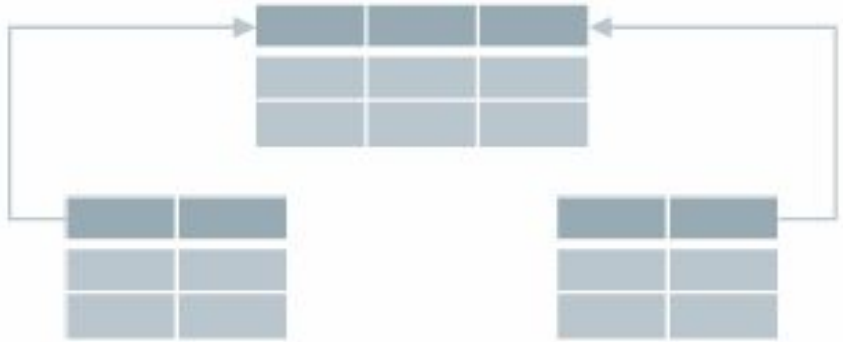
Source Systems



Source Systems - Databases

Databases

Relational Databases



NoSQL Databases

Key-Value

Document Stores

consuming data in the form of files, like text files, audio files like Mp3s, or even video or other types of files. While individual files like these might not sound like something you could call a source system, you'll find that in many cases, in your work as a data engineer, you need to download or be given access to files to begin your work.

Source Systems - Files

Files

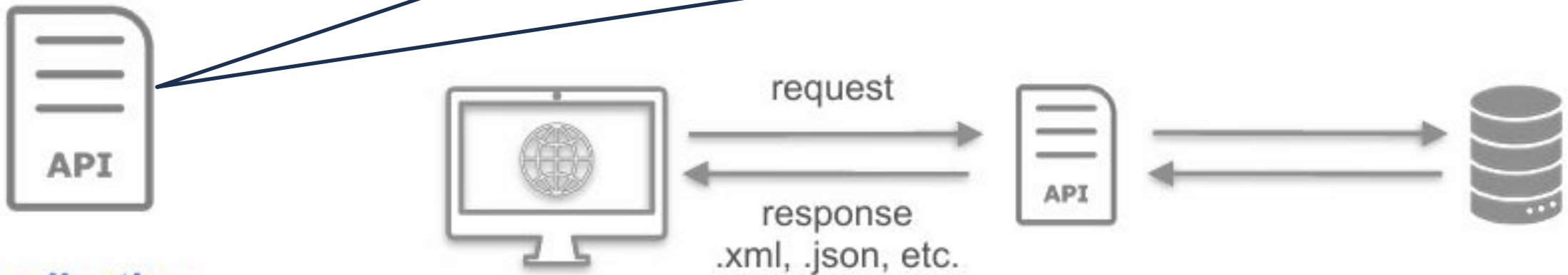
Text

Audio

Video

Source Systems - API

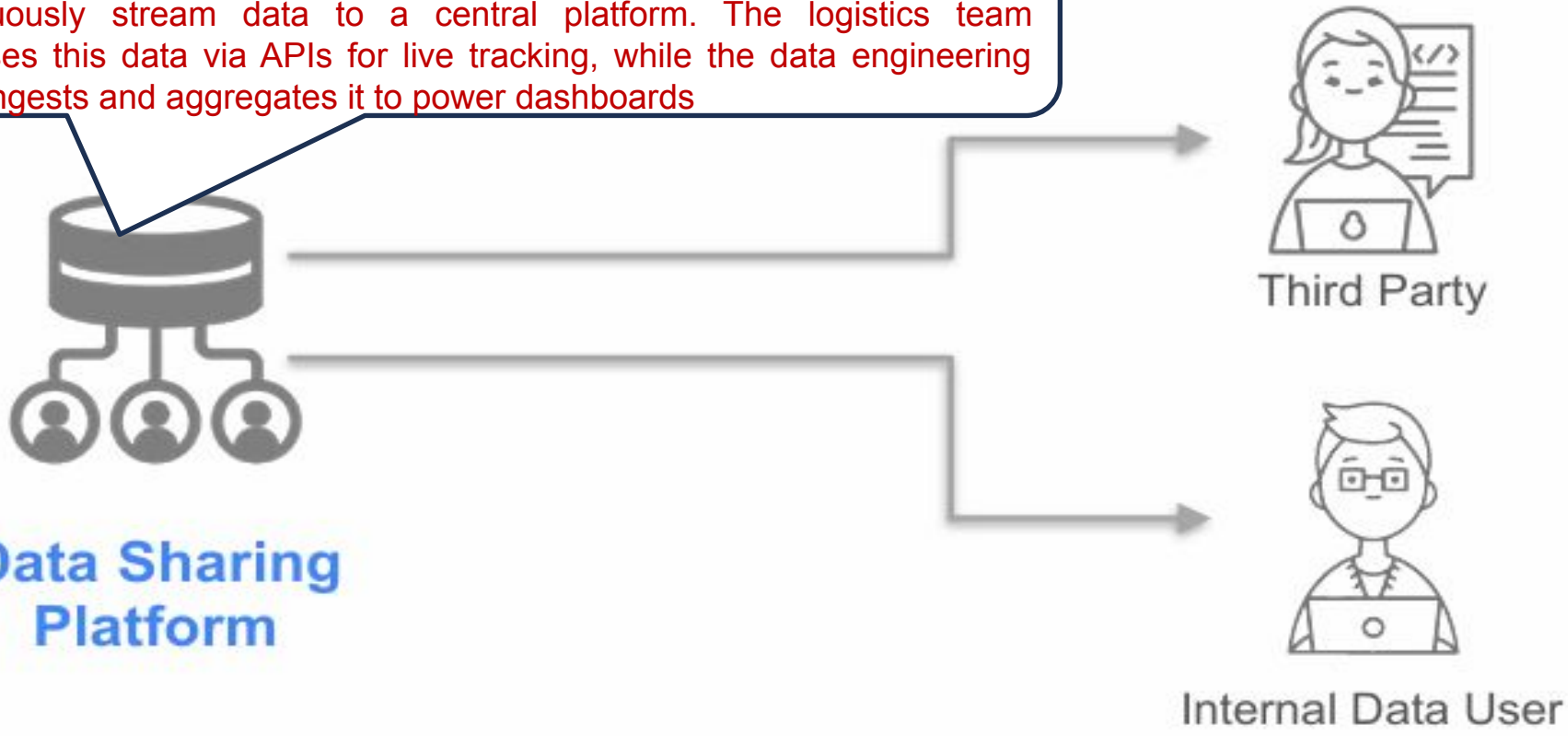
API allows you to issue a web request for data and then get that data back in a certain format, such as XML or JSON.



Application Programming Interface (API)

Source Systems - Data Sharing

In an e-commerce logistics network, each delivery truck may have IoT sensors tracking location, temperature, and door status. These sensors continuously stream data to a central platform. The logistics team accesses this data via APIs for live tracking, while the data engineering team ingests and aggregates it to power dashboards



Source Systems - IoT

IoT devices
Internet of Things

"Swarm" of IoT devices

Imagine your e-commerce platform depends on a shipping provider's API to get real-time tracking updates. One day, without notice, the provider changes a field name from status_code to shipment status, or starts using new status values like "DELAYED" instead of "LATE". If your downstream dashboards or alerting systems rely on the old format, they might break silently or misrepresent the data, unless your pipeline is built to detect and adapt to these changes.

Source Systems

Deliver data

Downstream Systems

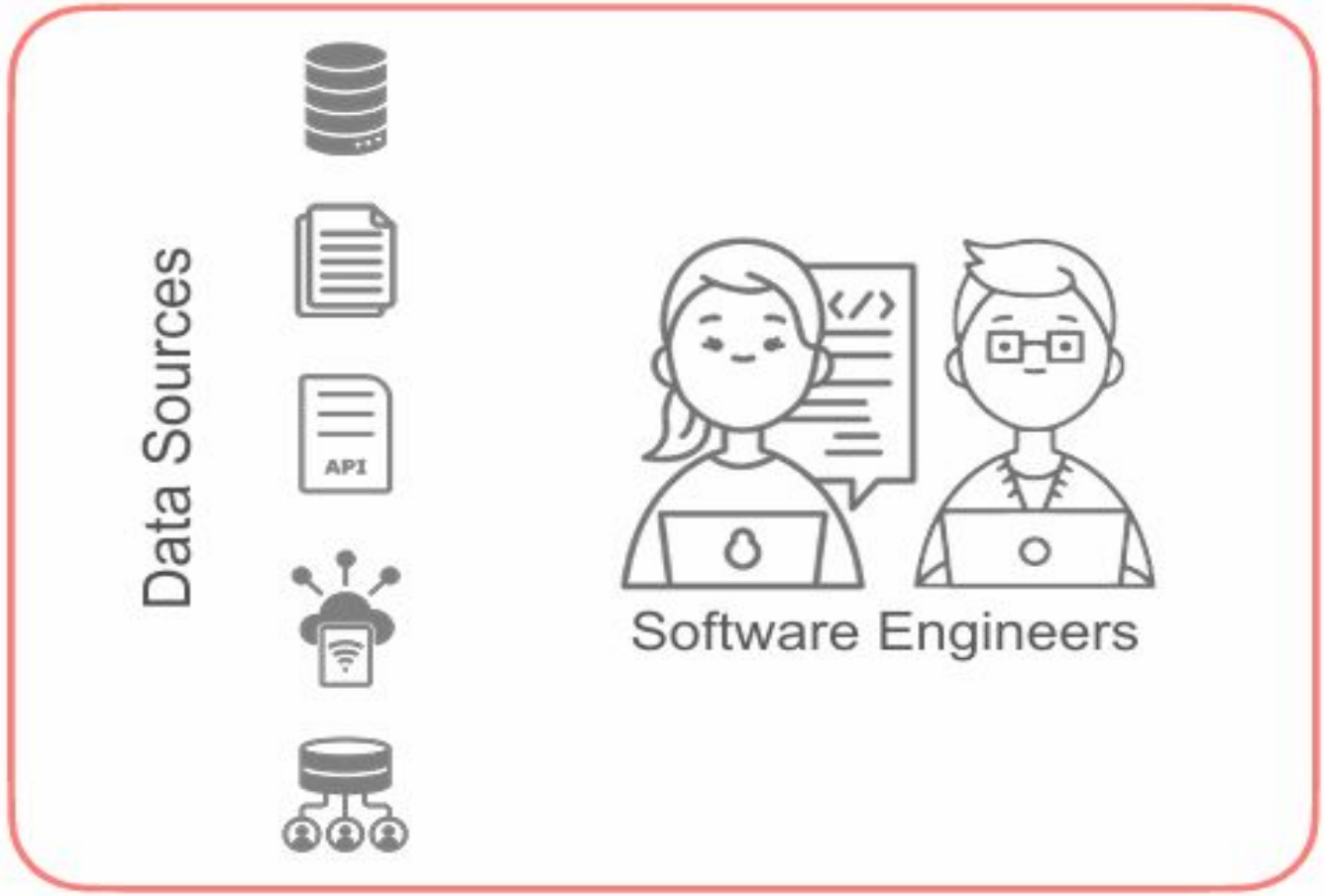
Unpredictable systems

- Systems go down
- Change in format/schema of data
- Change in data

Source Systems

- How are the systems set up?
- What kind of changes are to expect?

Source Systems



Data Engineer

Understand how source systems work

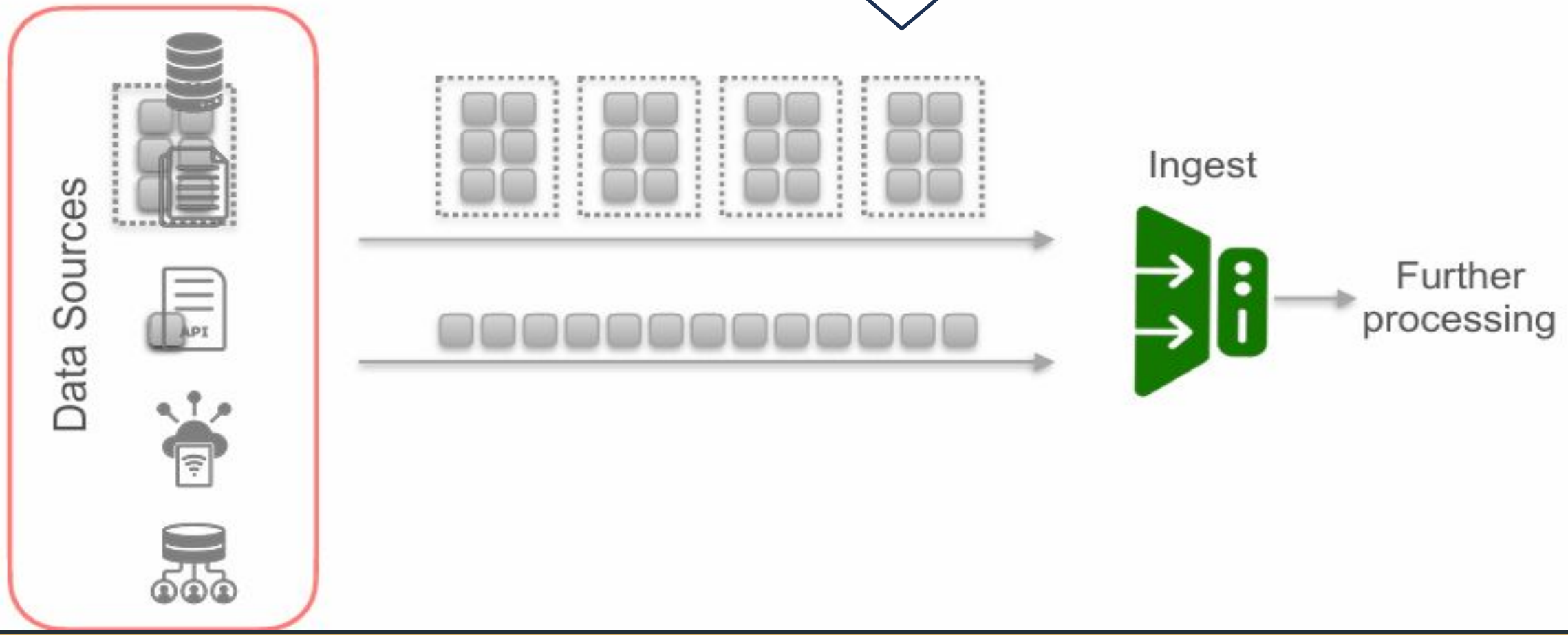
- How they generate data
- How the data may change over time
- How the changes will impact downstream systems

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Ingestion

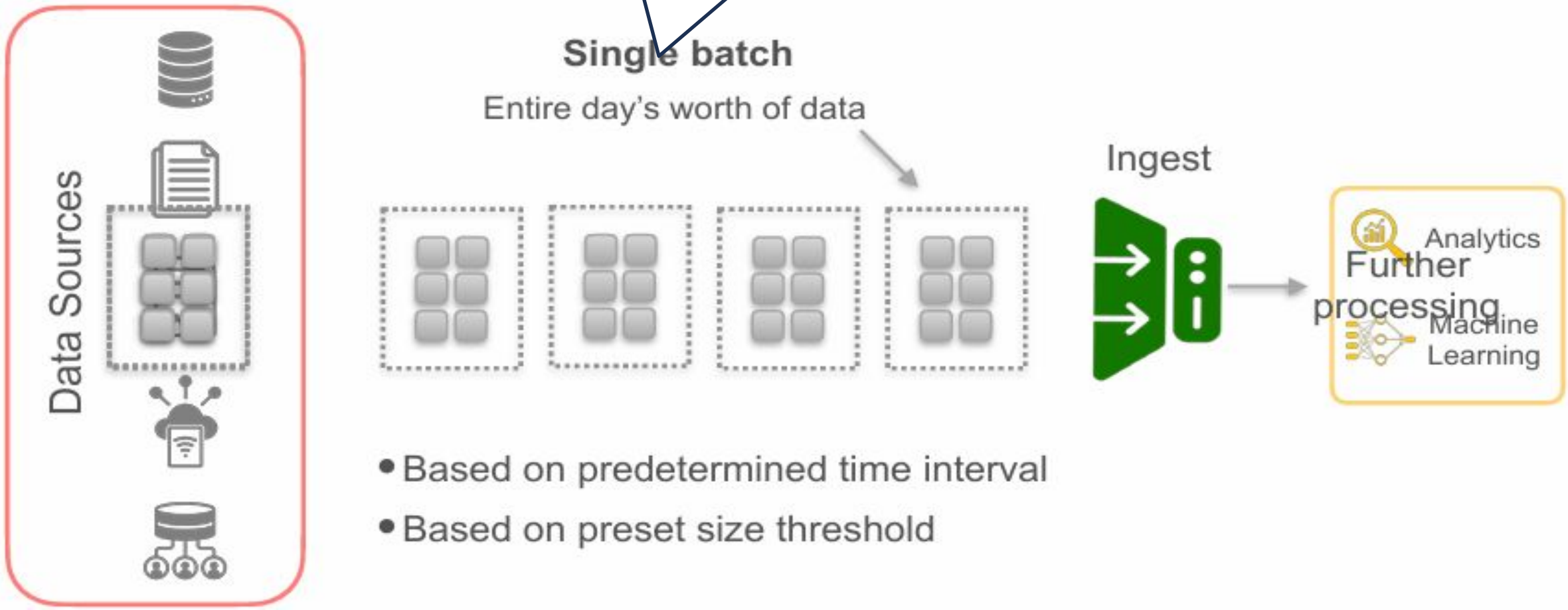
One of the most critical decisions in designing a data ingestion pipeline is determining how frequently data should be ingested from source systems into your data platform. This decision directly affects system architecture, performance, and the freshness of downstream analytics or applications.

Frequency of Ingestion



In batch ingestion, data is collected and moved at regular intervals—for example, every hour, every day, or based on a schedule.

Batch Ingestion



Streaming ingestion involves capturing and processing data as soon as it is generated, in near real-time or real time..

Streaming Ingestion

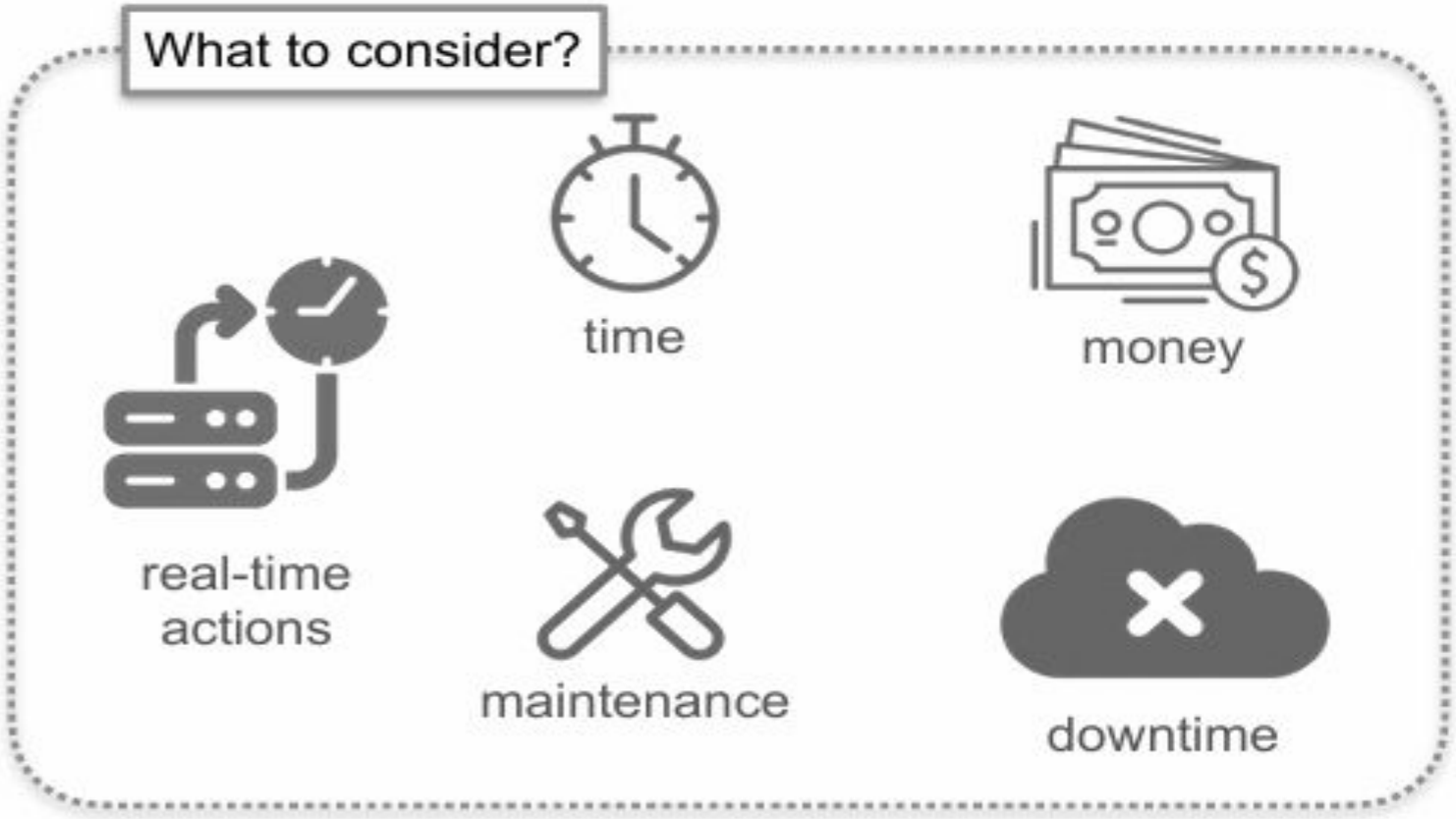
Batch ingestion collects and processes data at scheduled intervals (e.g., hourly or daily), making it suitable for non-real-time analysis. Streaming ingestion captures and processes data in real-time as events occur, ideal for time-sensitive applications like fraud detection or live dashboards.

Ingestion

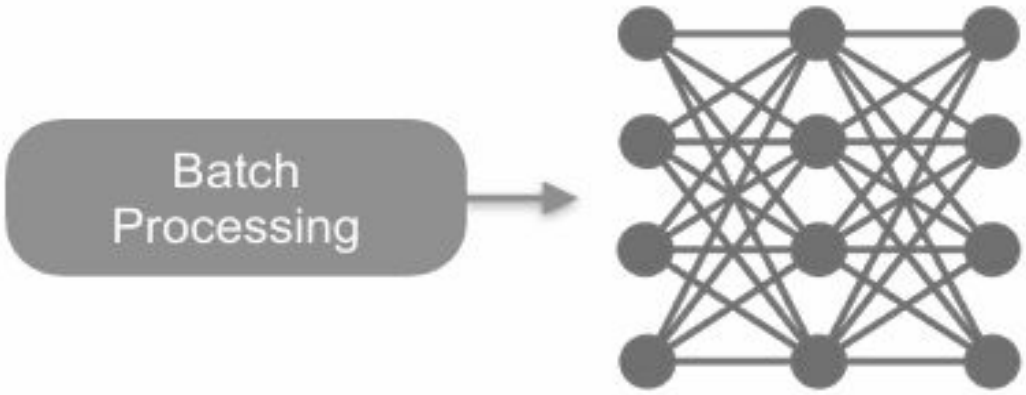
Batch Ingestion

vs.

Stream Ingestion

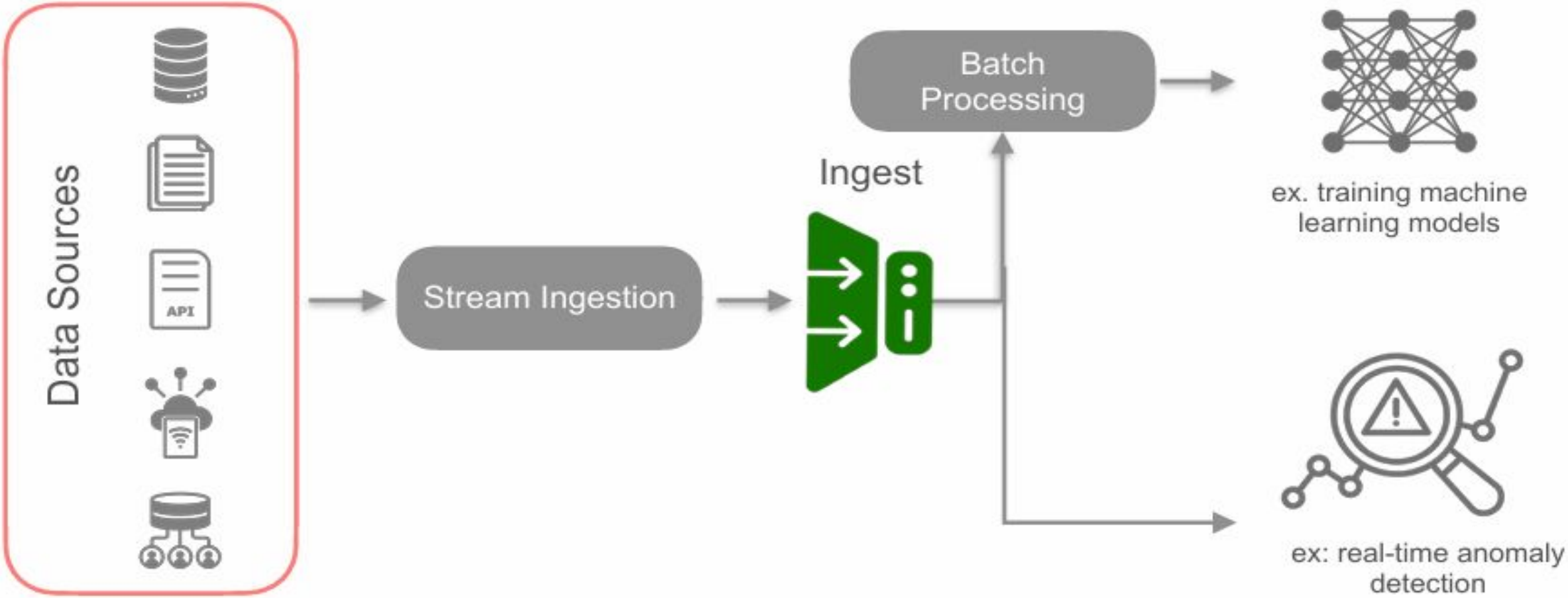


Streaming and Batch Components



ex. training machine learning models

Streaming and Batch Components



RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Storage

As a data engineer, the choice of storage solutions significantly impacts the functionality, performance, and cost of your systems. Storage begins with physical components like magnetic disks (still widely used due to low cost), solid-state drives (SSDs), and RAM (fast but expensive and volatile).

Raw Hardware Ingredients

Solid-state storage

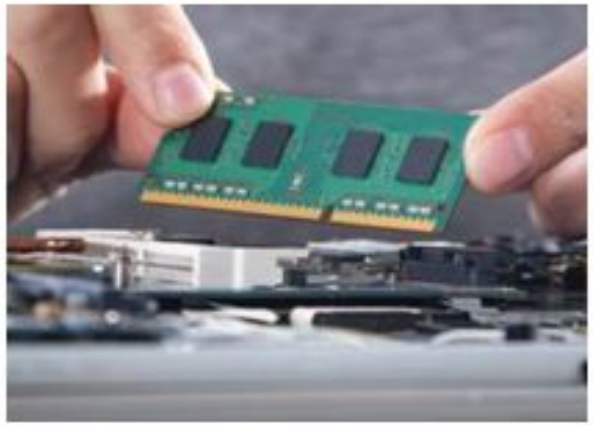
Magnetic disk

Magnetic disk

- Backbone of modern data storage system
- 2-3 times cheaper than solid-state storage

Raw Hardware Ingredients

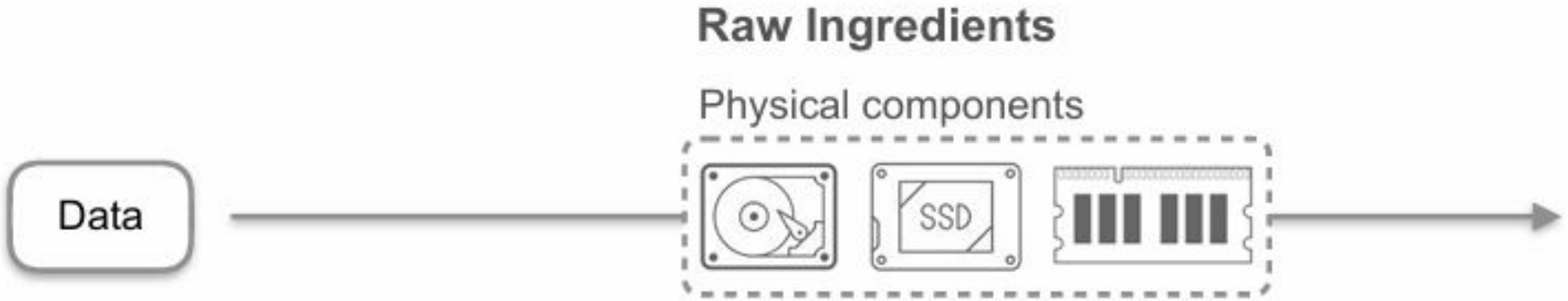
RAM (Random Access Memory)



RAM

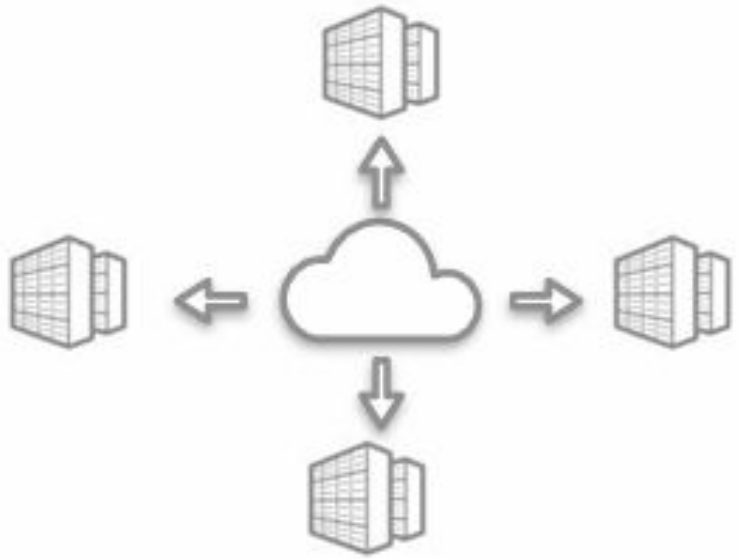
- Faster read and write speeds
- 30 - 50 times more expensive than solid-state storage
- Volatile

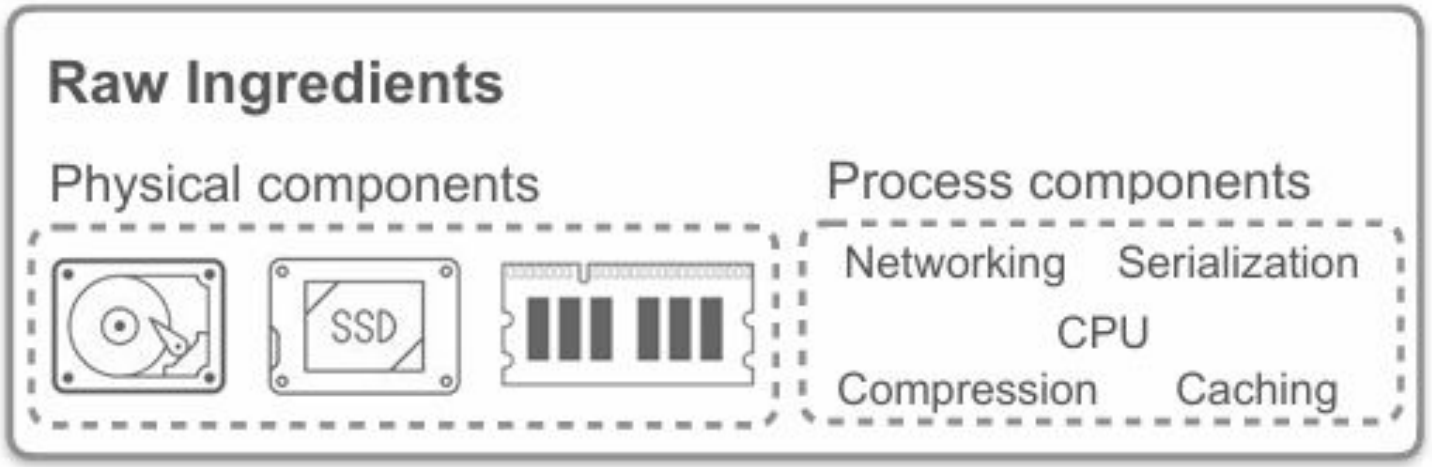
Storage



Modern and cloud-based storage systems are distributed across clusters and data centers, relying on essential components such as networking, CPUs, serialization, compression, and caching. While data engineers typically work with higher-level systems like databases, object storage (e.g., Amazon S3), and modern table formats (e.g., Apache Iceberg, Hudi), these systems are built on top of deeper hardware and software foundations.

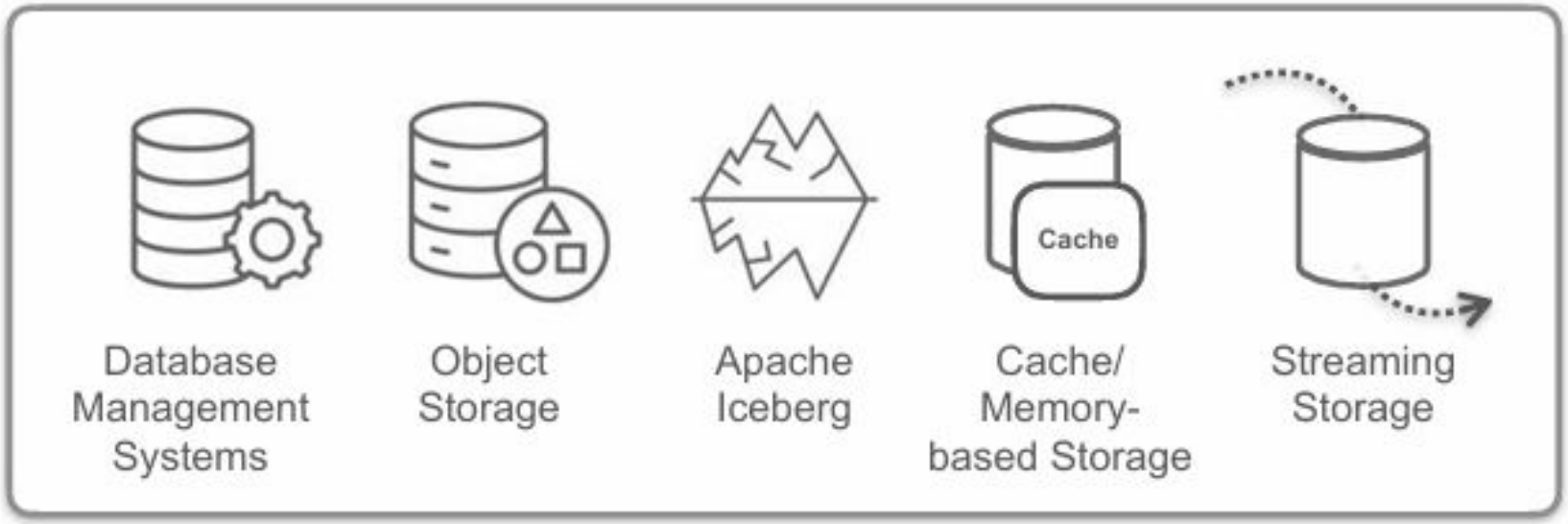
Storage





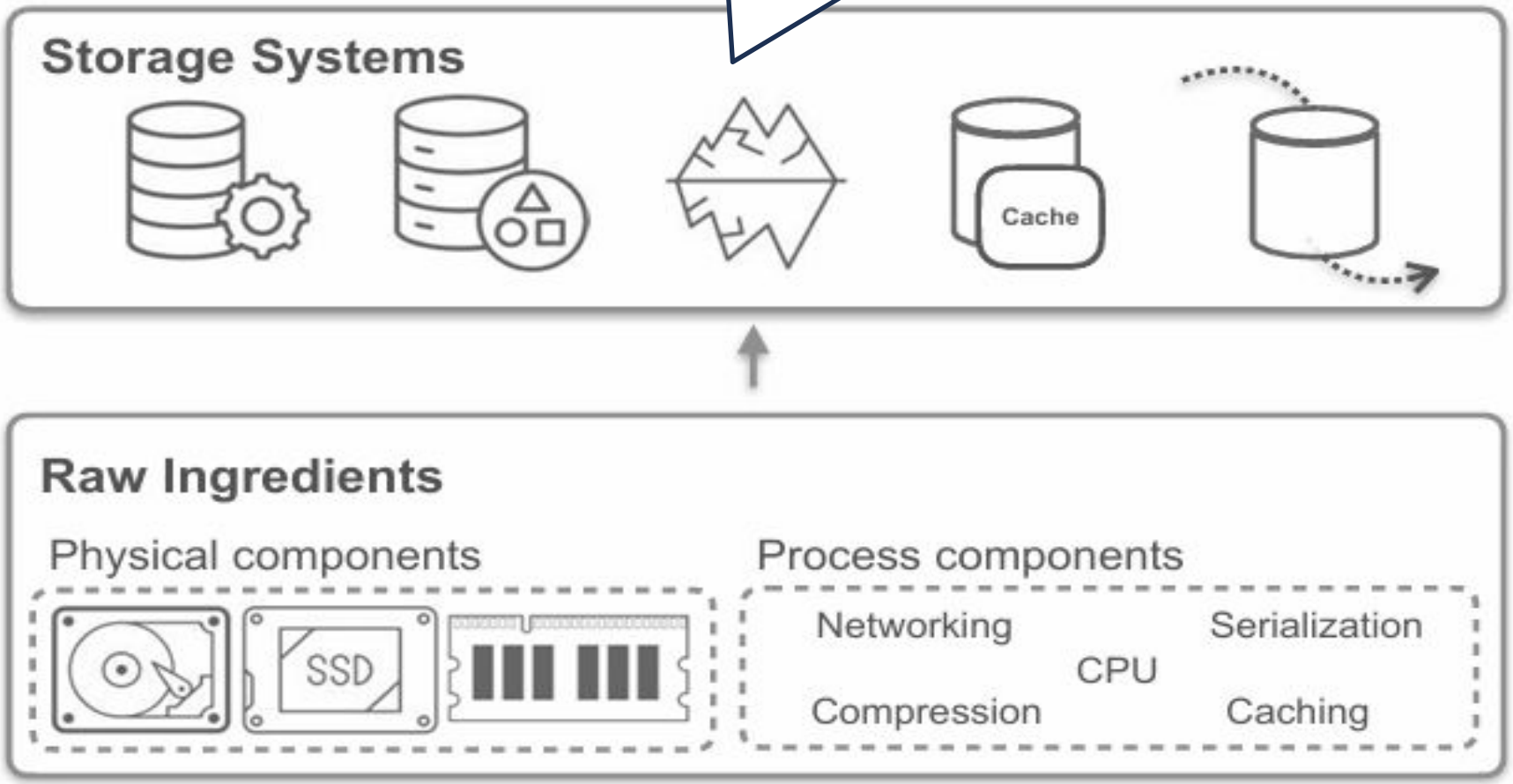
Data engineers typically work with higher-level systems like databases, object storage (e.g., Amazon S3), and modern table formats (e.g., Apache Iceberg, Hudi), these systems are built on top of deeper hardware and software foundations.

Storage Systems



Storage can be visualized as a three-layer hierarchy: Raw Ingredients – Physical (disks, RAM, SSDs) and non-physical (networking, protocols).Storage Systems – Tools like DBMS, object stores, and streaming storage.

Storage Systems



Storage Abstractions – High-level constructs like data warehouses, data lakes, and data lake houses, which combine systems for scalable and flexible storage solutions.

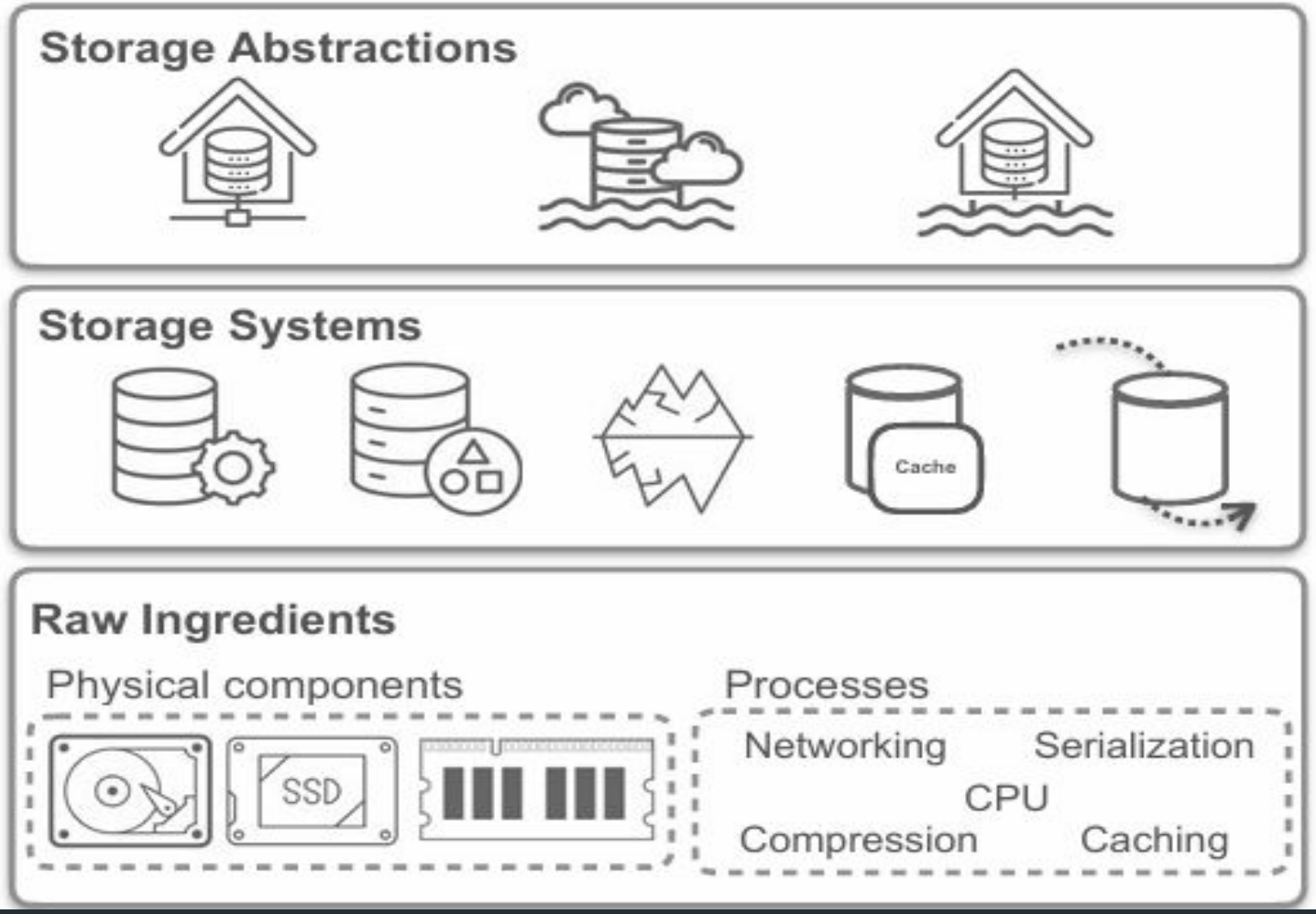
Storage Abstractions

Storage abstractions: combinations of storage systems

Choose configuration parameters:

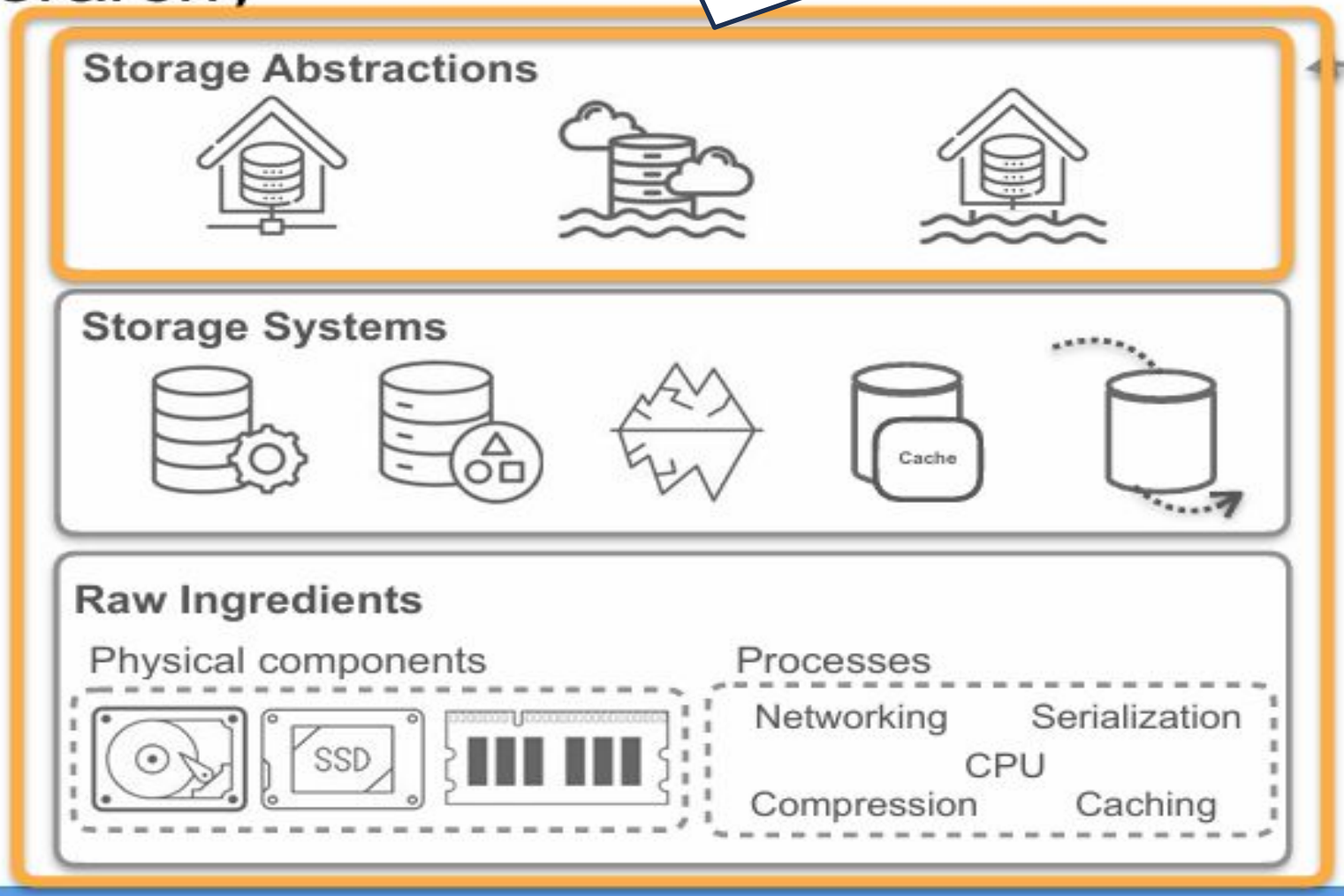
- Latency
- Scalability
- Cost

Storage Hierarchy



As a data engineer, you may mostly work at the abstraction level, configuring systems for latency, scalability, and cost—but having a deep understanding of the entire hierarchy helps avoid costly mistakes.

Storage Hierarchy



Work near or at the top

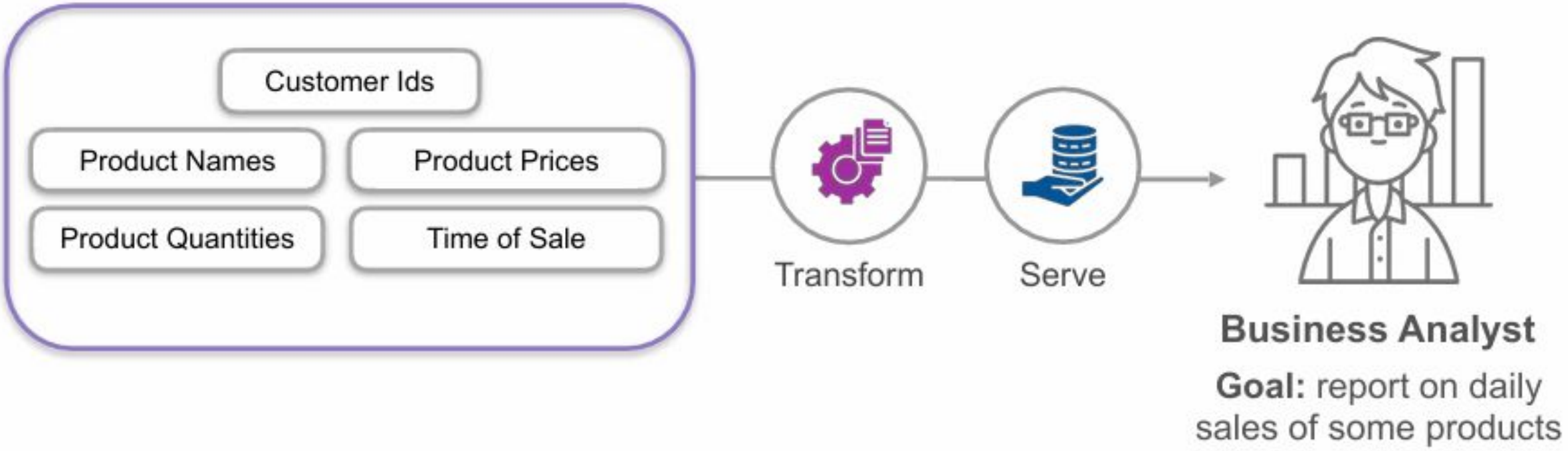
Understand the details of your entire storage solution

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Queries, Modelling and Transformation

The **transformation stage** is where data engineers begin to **add real value**, by converting **raw ingested data** into formats that are **useful and consumable** by downstream users such as analysts and data scientists. **transformation stage** is where data engineers begin to **add real value**, by converting **raw ingested data** into formats that are **useful and consumable** by downstream users such as analysts and data scientists.

Transformation



Transformation

Query

Issuing a request to read records from a database or other storage system.

Data Warehouse

Query

- Tabular data
- Semi-structured data

Query

Issuing a request to read records from a database or other storage system.

Query Language

SQL Commands

Data Cleaning

DROP
TRUNCATE
TRIM
REPLACE
SELECT DISTINCT

Data Joining

INNER JOIN
LEFT JOIN
RIGHT JOIN
FULL JOIN
UNION

Data Aggregating

SUM
AVG
COUNT
MAX
MIN
GROUP BY

Data Filtering

WHERE
AND
OR
IS NULL
IS NOT NULL
IN
LIKE

Query

Issuing a request to read records from a database or other storage system.

Poor queries: negative impact on the source database

Source Database

Query

Issuing a request to read records from a database or other storage system.

Poor queries: cause row explosion in your database

Query

You database

"Row explosion" in the context of databases and data analysis refers to a situation where a join operation produces an unexpectedly large number of rows, often due to a flawed join condition or the absence of a join condition entirely.

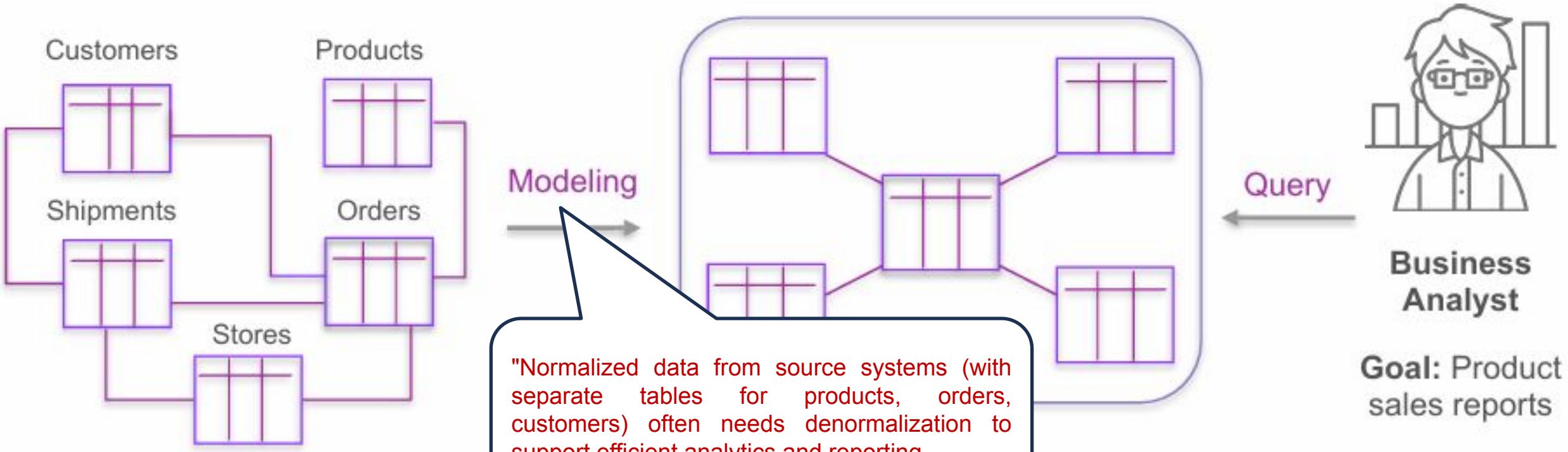
Query

Issuing a request to read records from a database or other storage system.

Poor queries: cause downstream delays

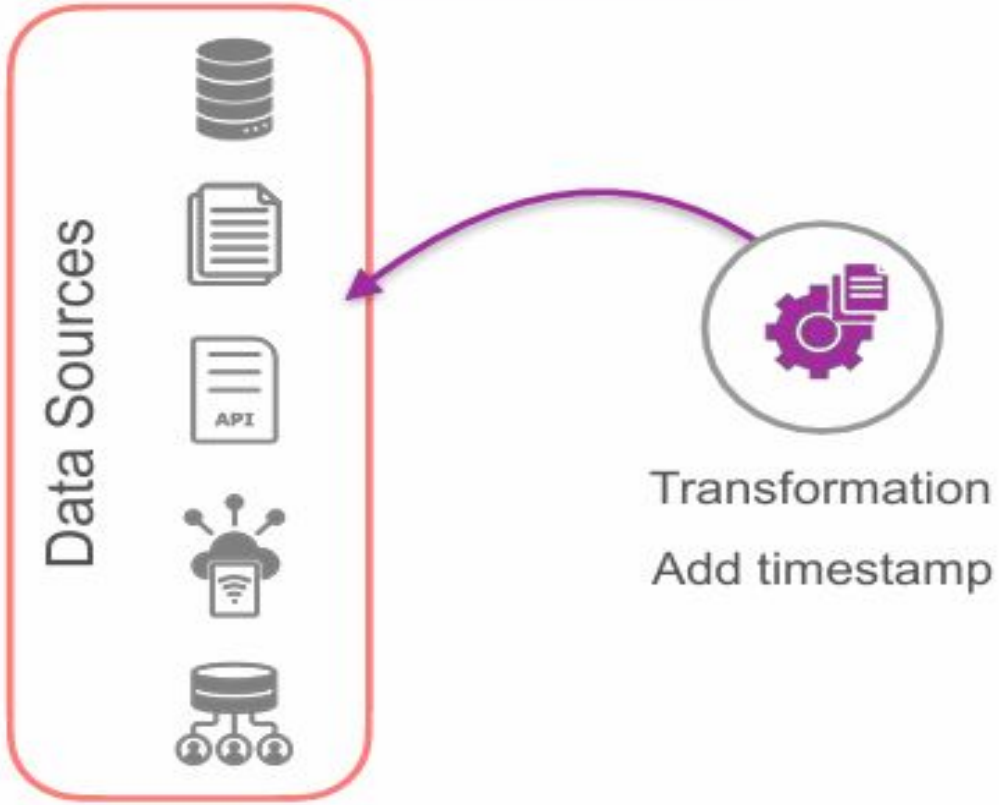
Data modeling

Choosing a coherent structure for your data to make it useful for the business.



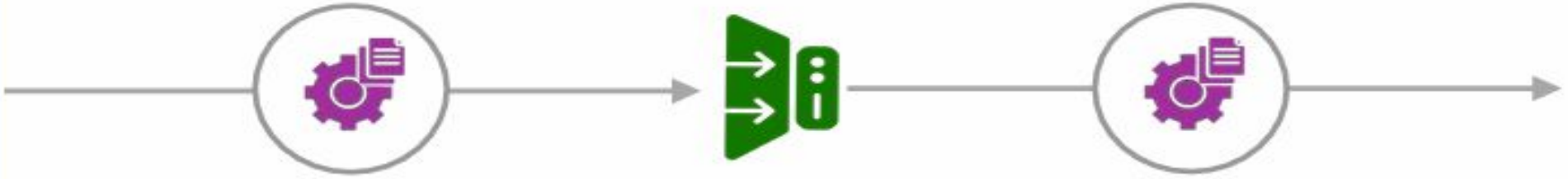
Data transformation

Data manipulated, enhanced and saved for downstream use.



Data transformation

Data manipulated, enhanced and saved for downstream use.



Transformation

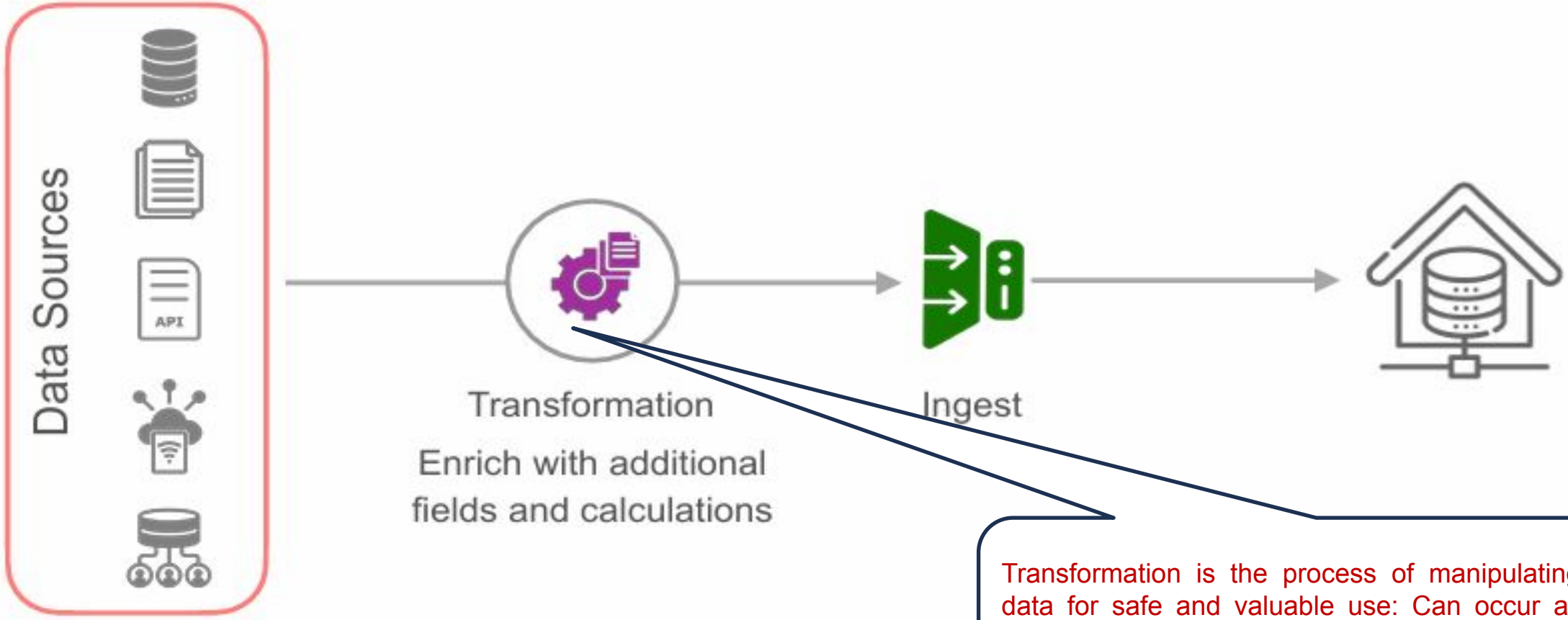
Ingest

Transformation

- map data into correct types
- put records into standard formats

Data transformation

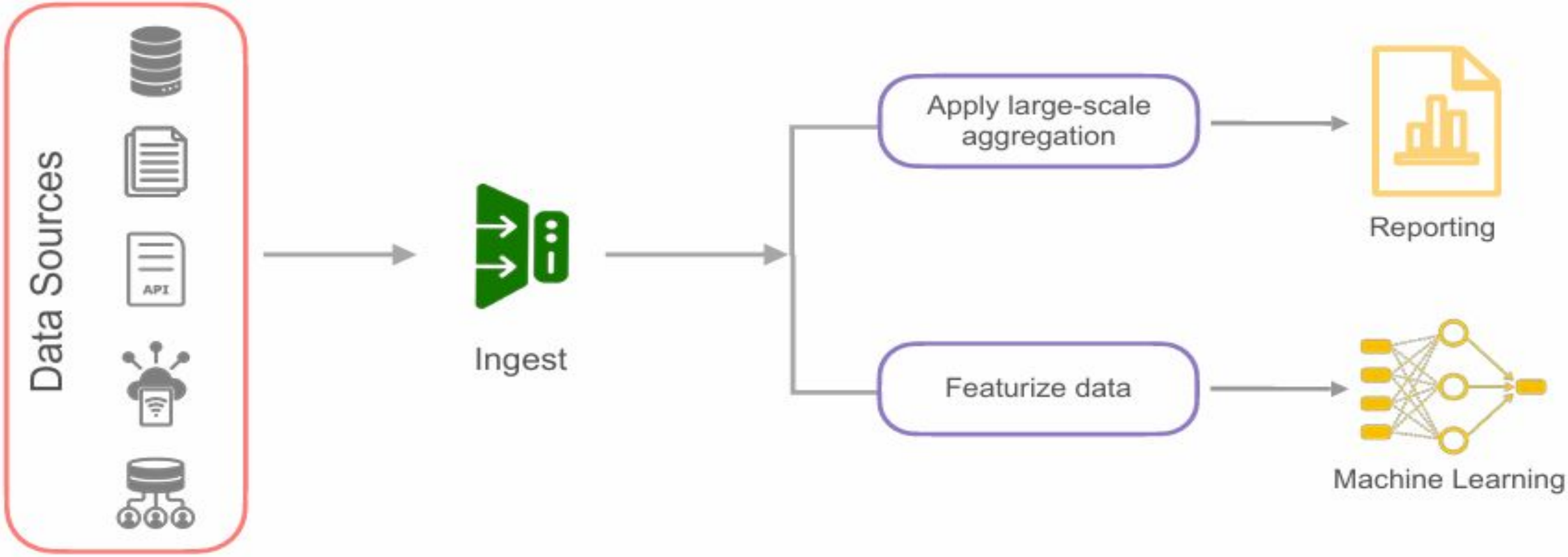
Data manipulated, enhanced and saved for downstream use.



Transformation is the process of manipulating and enriching data for safe and valuable use: Can occur at source, during ingestion, or post-ingestion Includes tasks like type mapping, schema alignment, aggregations, denormalization, and feature

Data transformation

Data manipulated, enhanced and saved for downstream use.

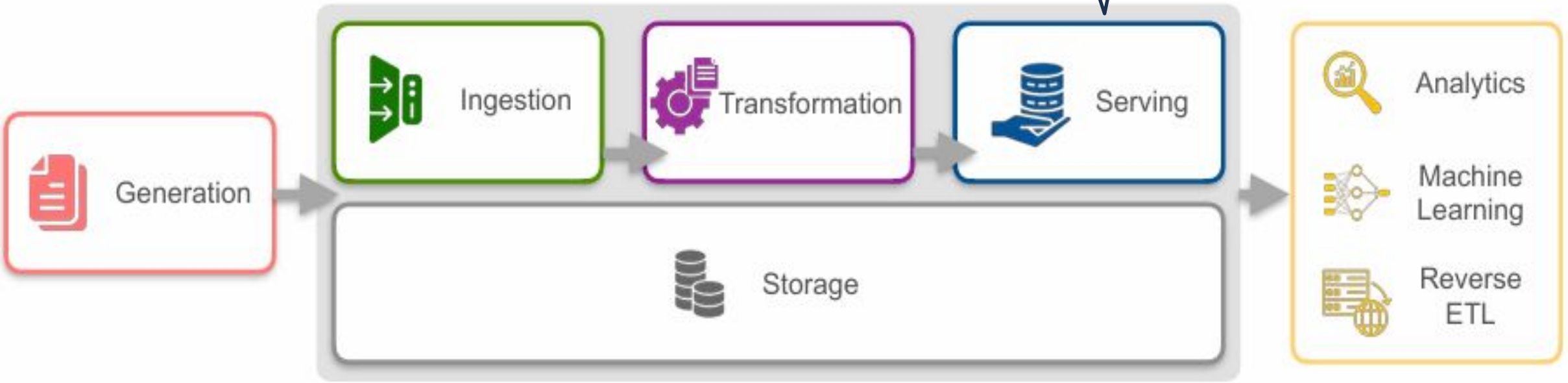


RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Serving Data

it's about delivering it in the right form, at the right time, to the right users

The Data Engineering Lifecycle



Analytics

Business Intelligence (BI): Analysts use dashboards and reports to explore historical data and derive insights (e.g., marketing performance, customer experience).Operational Analytics: Real-time monitoring for immediate action (e.g., tracking website uptime using log data).Embedded Analytics: External, user-facing data insights (e.g., banking dashboards, smart thermostat apps).

Analytics is the process of identifying key insights and patterns within data.

Business Intelligence

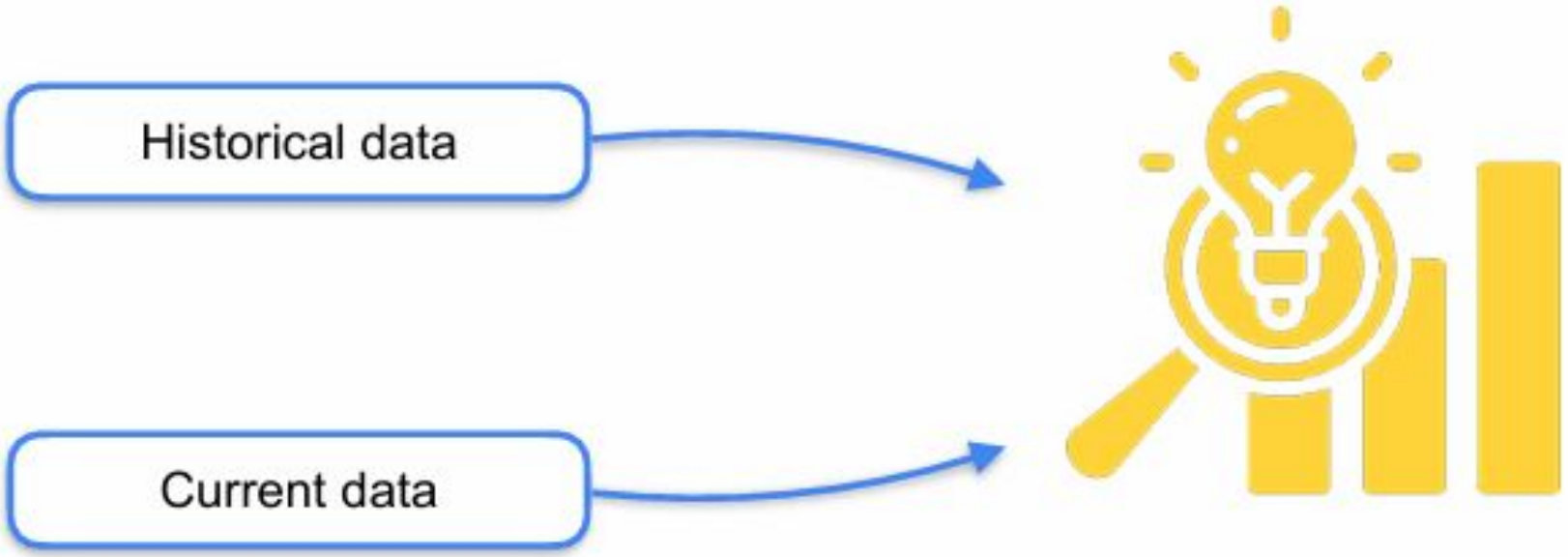
Operational Analytics

Embedded Analytics

Analytics

Business Intelligence

Explore historical and current business data to discover insights



Analytics

Business Intelligence

Explore historical and current business data to discover insights

Analytics

Business Intelligence

Explore historical and current business data to discover insights

Reports

Dashboards

Spot patterns and trends

Monitor:

- Campaign engagement
- Regional sales
- Customer experience metrics

Analytics

Business Intelligence

Explore historical and current business data to discover insights

Analytics

Business Intelligence

Explore historical and current business data to discover insights

Analyst pulls more data

SQL

select .. from table …

Analytics

Real-time monitoring for immediate action (e.g., tracking website uptime using log data).

Operational Analytics

Monitoring real-time data for immediate action

E-commerce website

Real-time website performance metrics

Analytics

Operational Analytics

Monitoring real-time data for immediate action

Ingest and transform data (from sensor, etc.)Store data in systems optimized for fast access Serve it in a format usable by front-end apps (often via APIs or dashboards)

Analytics

Embedded Analytics

External or customer-facing analytics

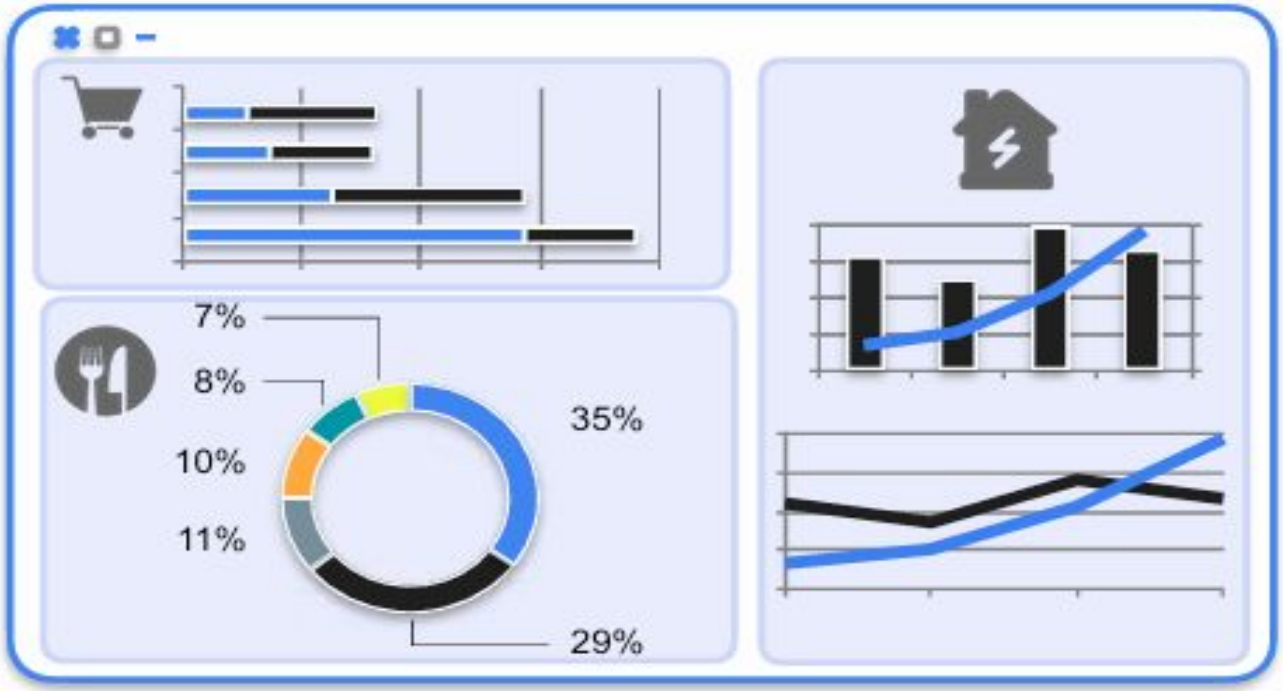
Ingest and transform data (from transactions, etc.)Store data in systems optimized for fast access Serve it in a format usable by front-end apps (often via APIs or dashboards)

Analytics

Embedded Analytics

External or customer-facing analytics

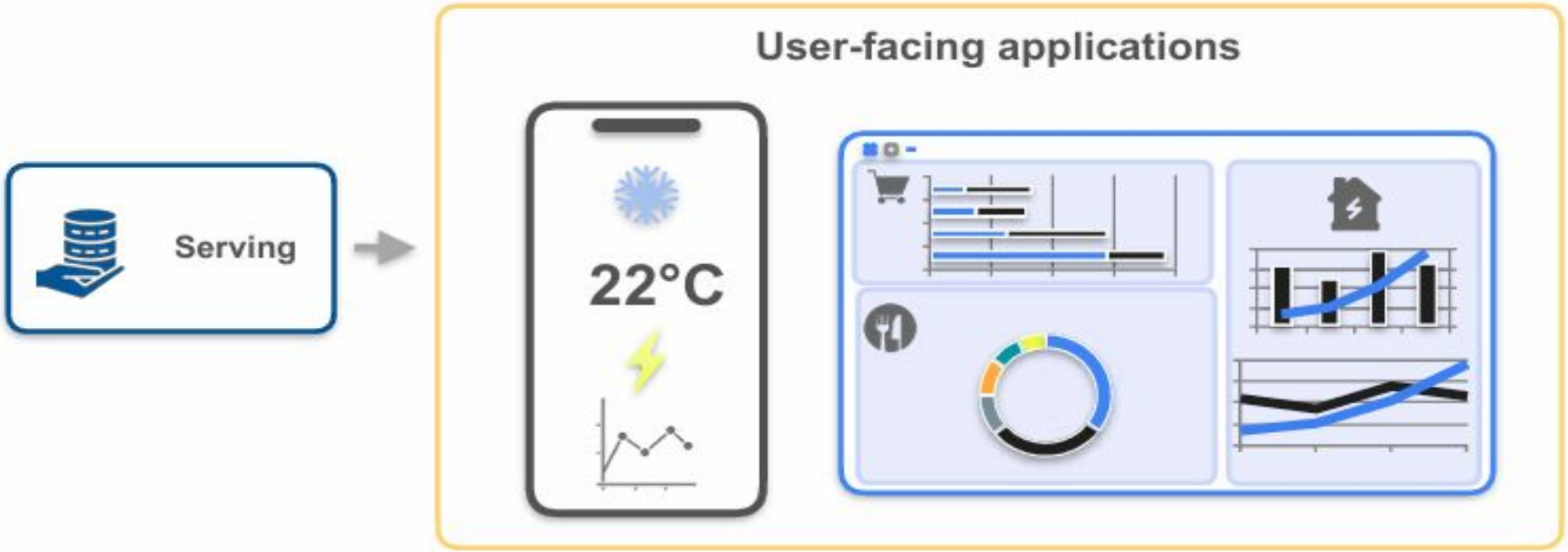
Customer-facing dashboards



Analytics

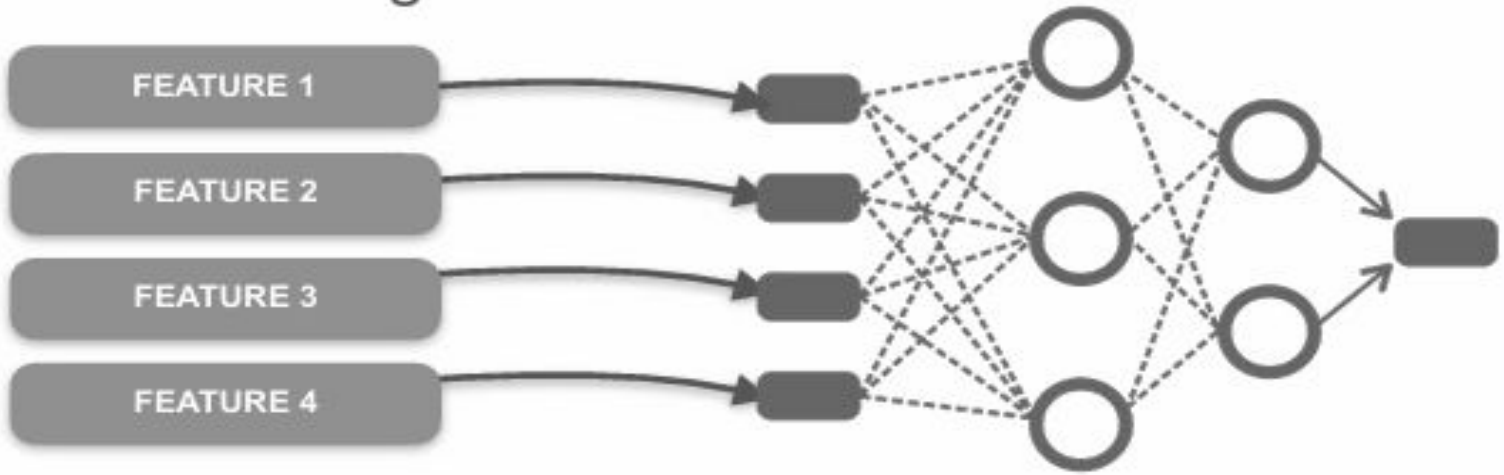
Embedded Analytics

External or customer-facing analytics



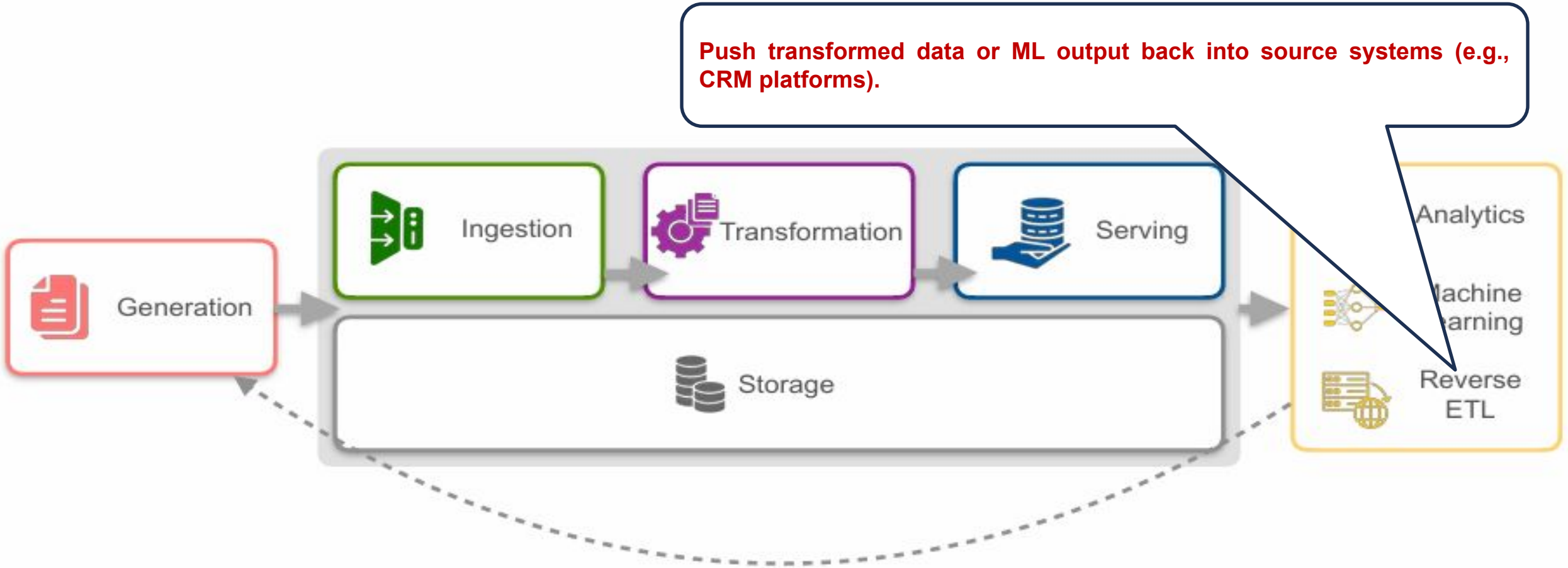
Machine Learning

- Model training



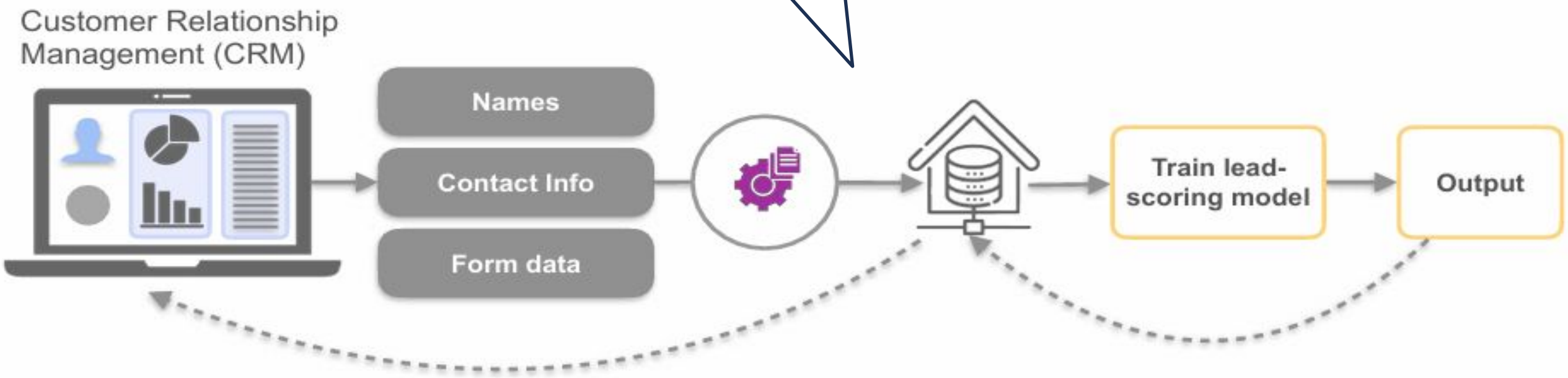
- Real-time inference
- Track data history and lineage

The Data Engineering Lifecycle



Push transformed data or ML output back into Datawarehouse and CRM

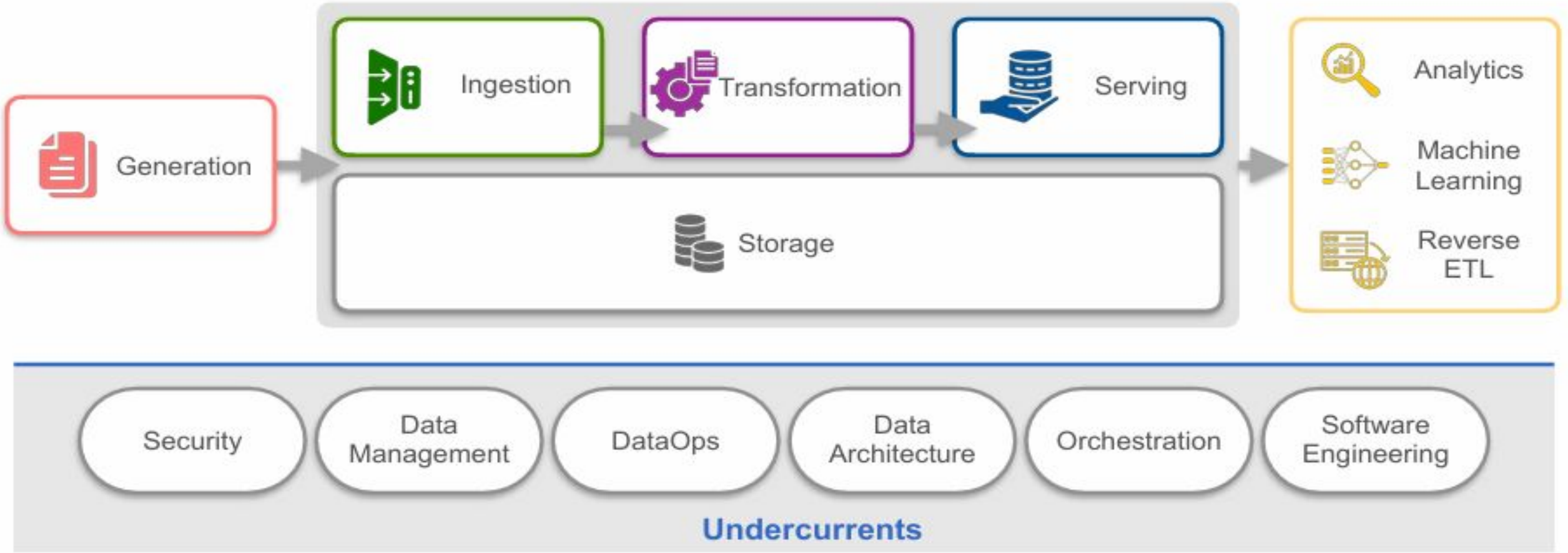
Reverse ETL



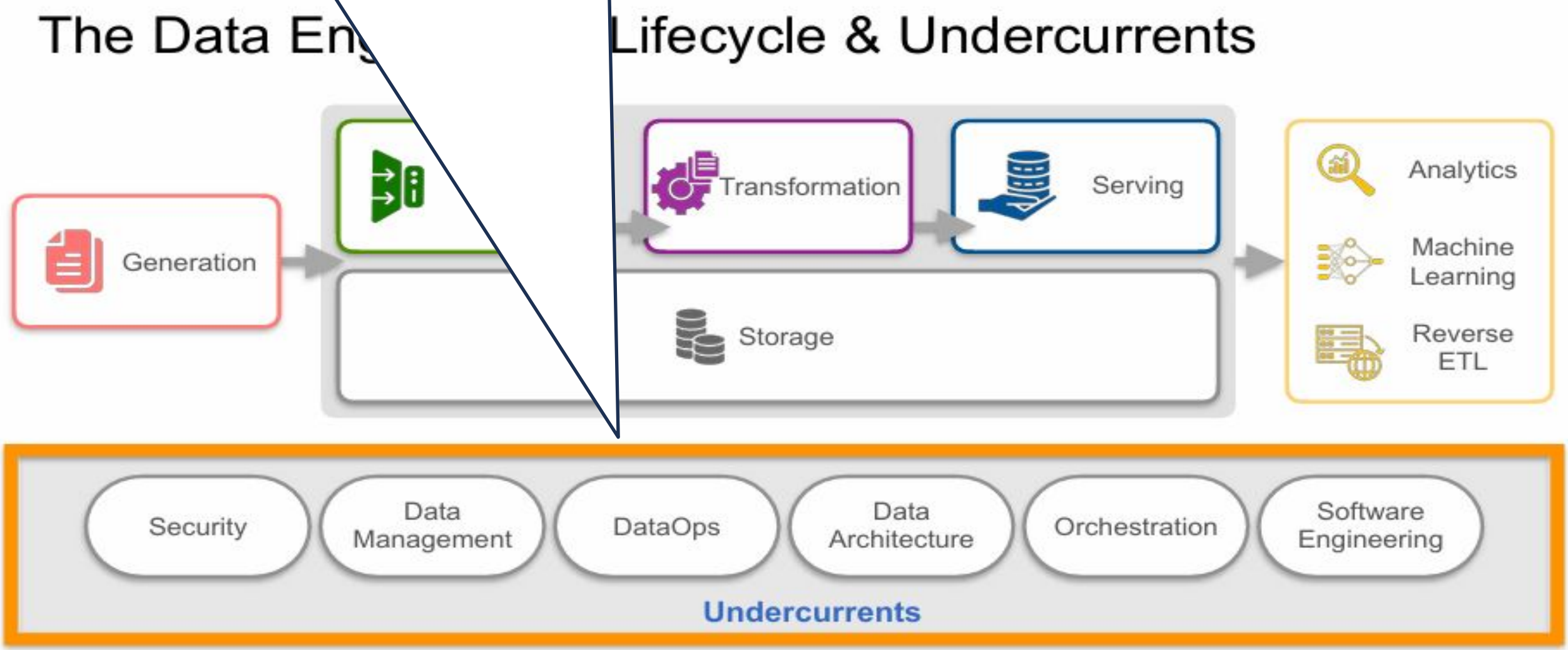
RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Intro to the Undercurrents

The Data Engineering Lifecycle & Undercurrents



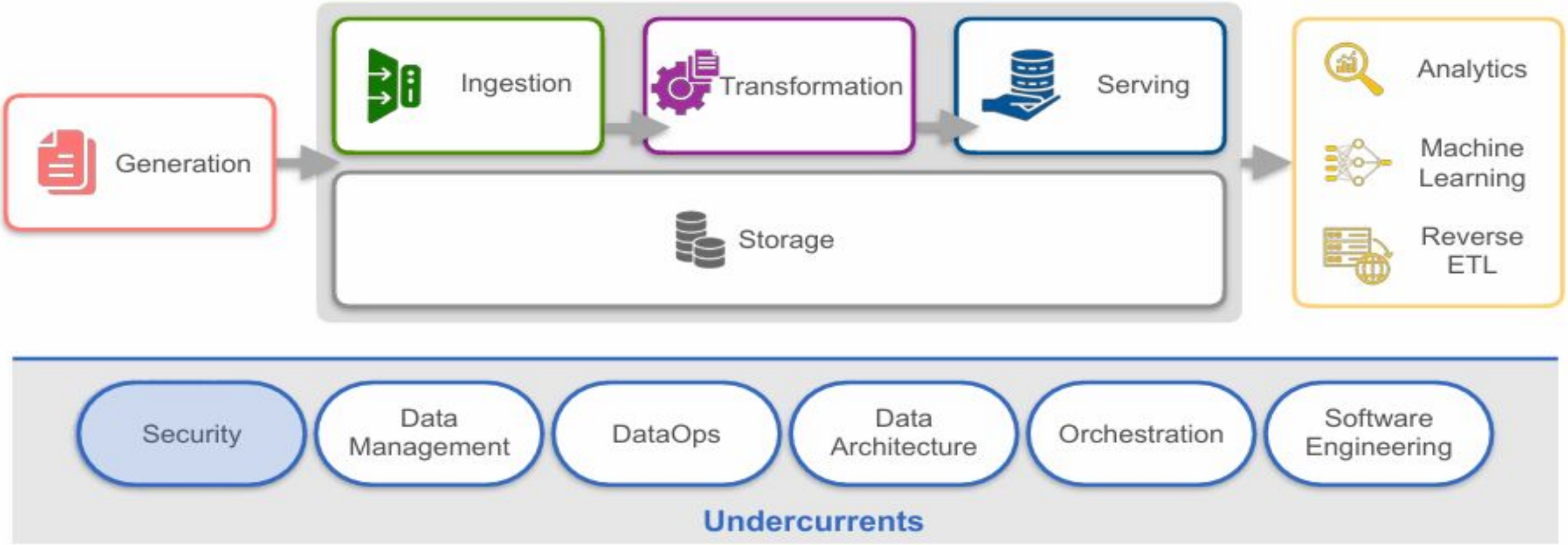
In data engineering, "undercurrents" refer to the foundational, cross-cutting concerns that span across all stages of the data lifecycle — from ingestion to transformation, storage, and serving. These aren't separate phases, but rather critical background elements that influence the quality, reliability, and performance of your entire data system.



RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

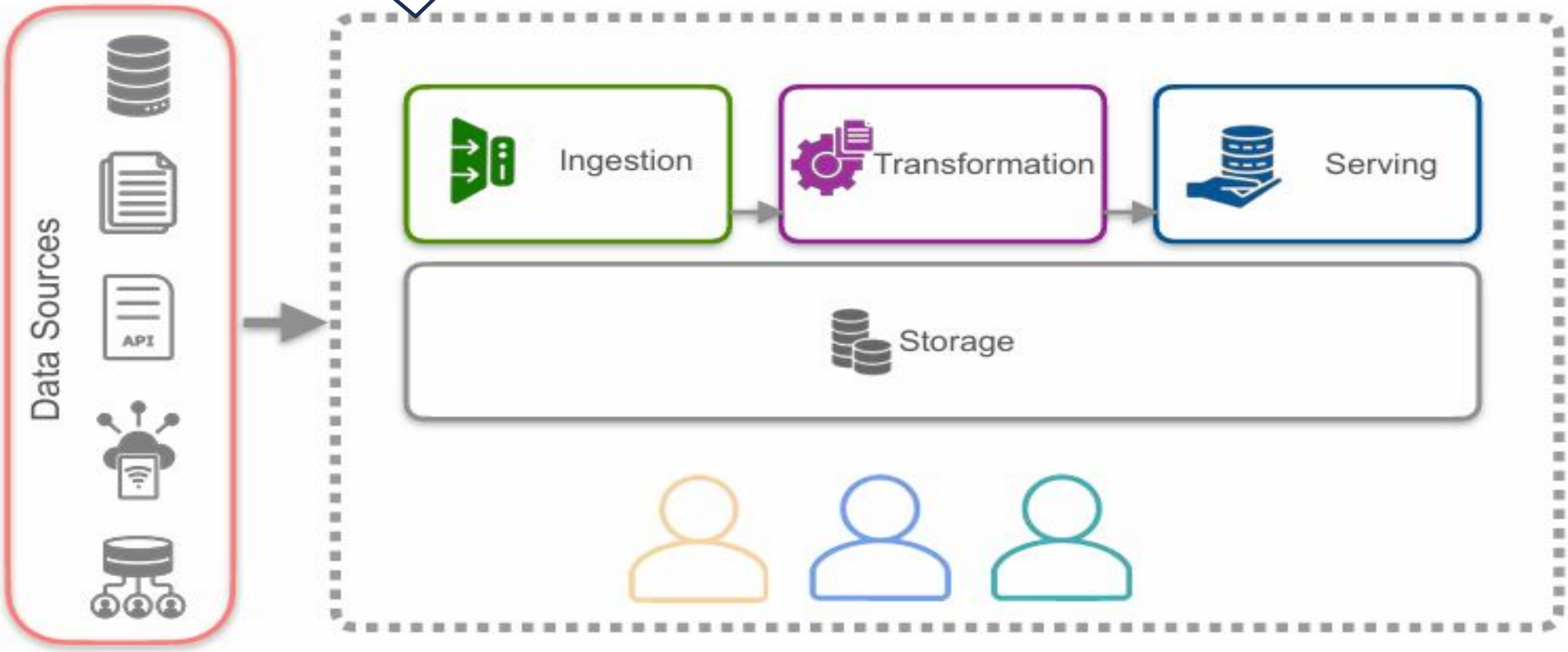
Security

The Data Engineering Lifecycle & Undercurrents

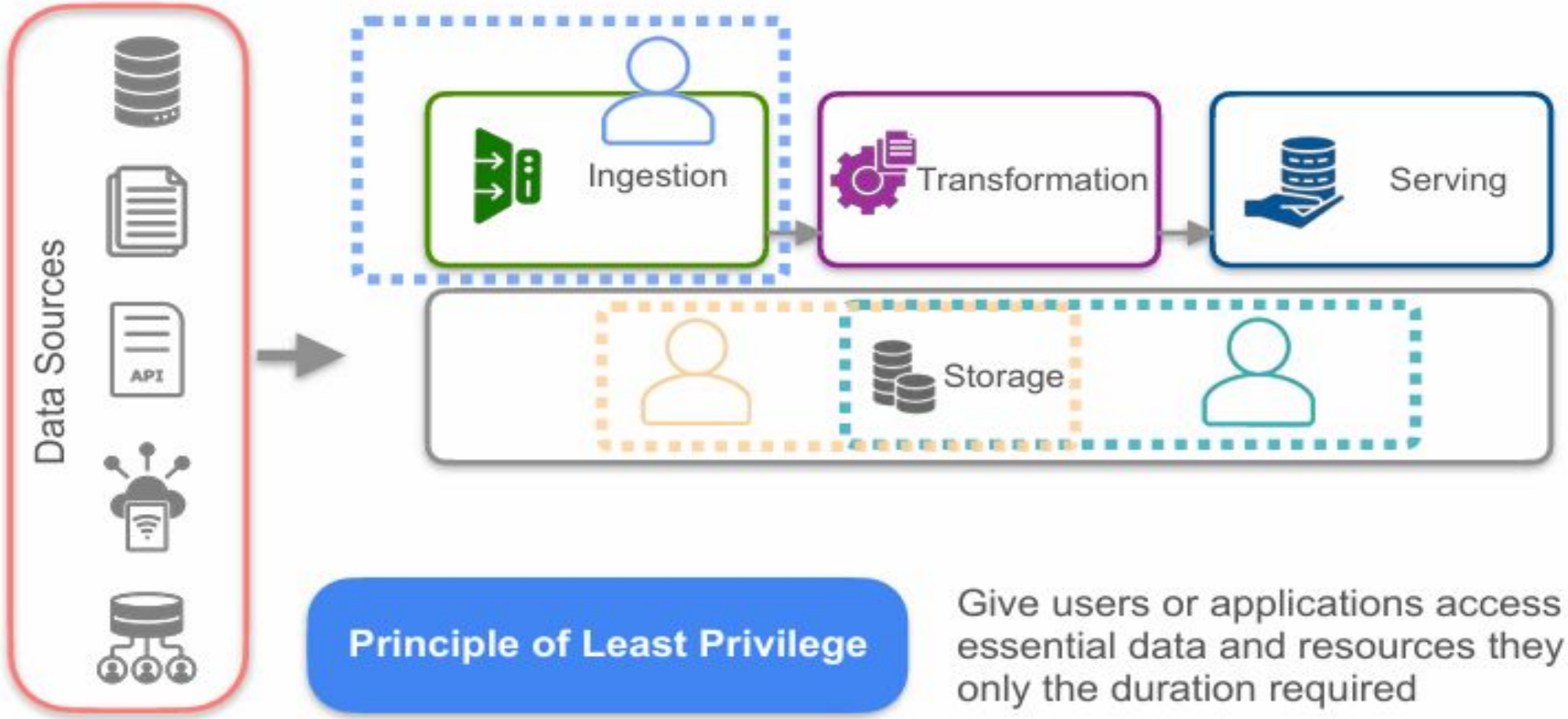


Data engineers play a critical role in enforcing security across the data pipeline by combining principles, protocols, and cultural practices.

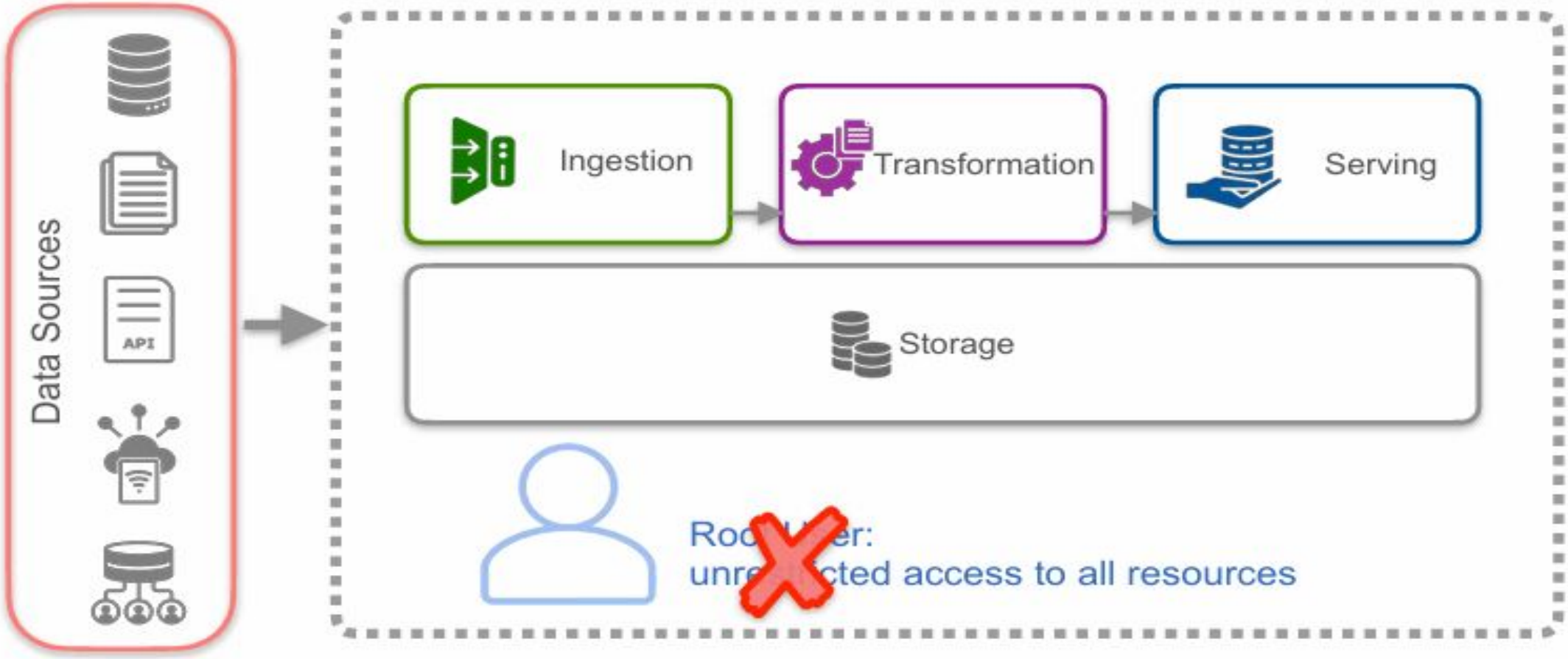
Security



Security


Data Sources
API
Ingestion
Transformation
Serving
Storage
Principle of Least Privilege
Give users or applications access to only the essential data and resources they need for only the duration required

Security


Data Sources
API
Ingestion
Transformation
Serving
Storage
unrestricted access to all resources

Restrict access to sensitive data unless absolutely necessary.
If sensitive data isn't needed, don't ingest or store it—eliminates the risk entirely.

Data Sensitivity

Id	First Name	Last Name	Credit Card Number
25	John	Smith	457893
45	Lara	Jones	347891

Data sensitivity

Id	First Name	Last Name	Credit Card Number
25	J****	S****	****93
45	L****	J****	****91

Security in the Cloud

- Identity and Access Management (IAM)
- Encryption Methods
- Networking Protocols

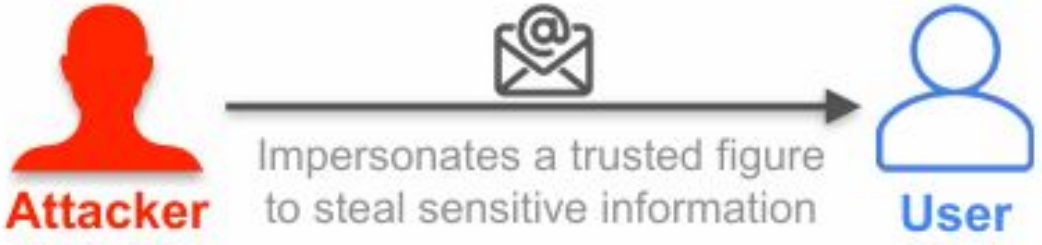
Many data breaches occur due to: Sharing passwords insecurely Falling for phishing scams ,Exposing cloud storage (e.g., S3 buckets) unintentionally

Security

Defensive Mindset

Be cautious with sensitive data

Design for potential attacks



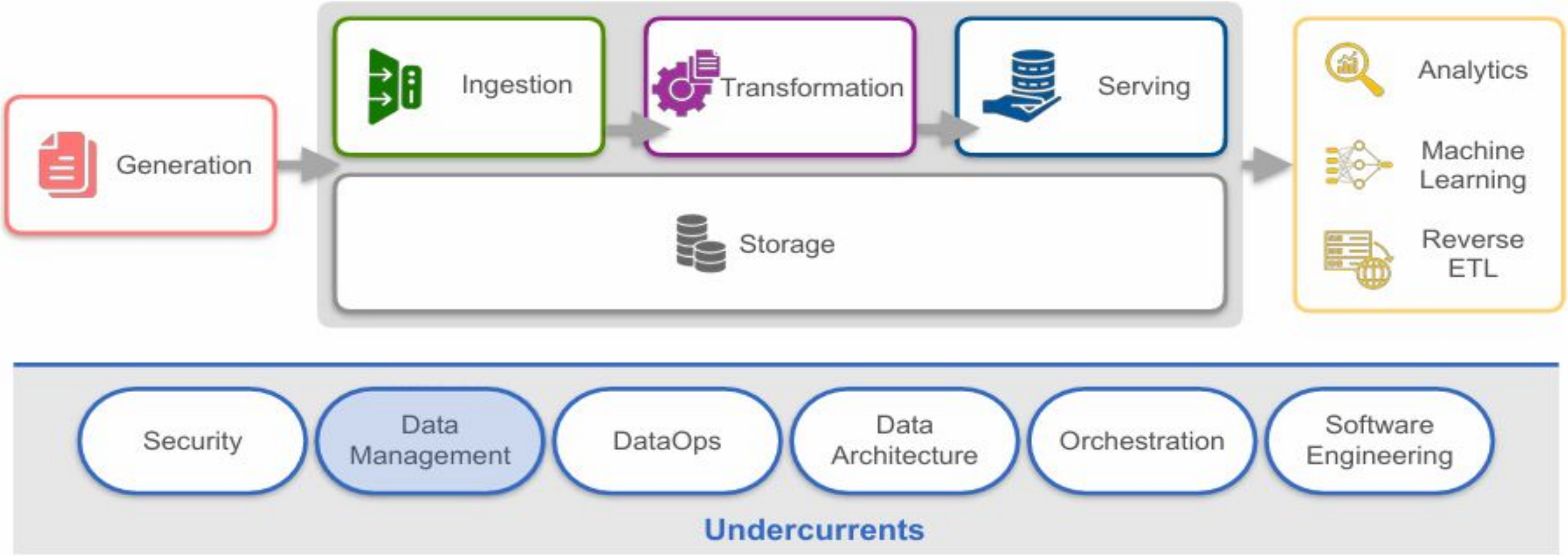
Security should be a mindset, not a checklist. A strong security culture focuses on understanding threats, minimizing risk, and making secure behavior a habit—not just putting on a show.

Security

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Data Management

The Data Engineering Lifecycle & Undercurrents



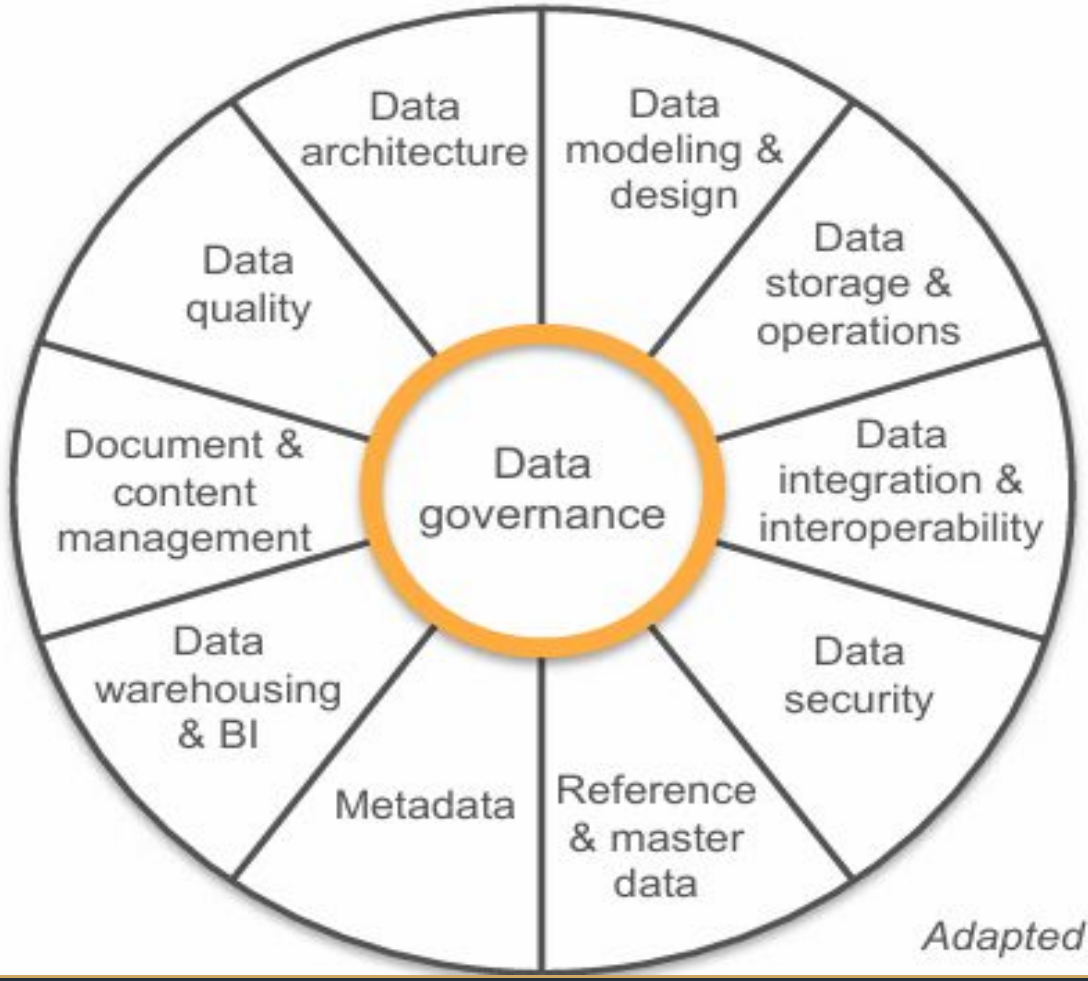
Data Management

"Data management is the development, execution, and supervision of plans, programs, and practices that deliver, control, protect, and enhance the value of data and information assets throughout their life cycles."

DMBOK's Definition

Data Management

11
Data Knowledge Areas



Adapted from: DAMA International

Data Governance

"Data governance is, first and foremost, a data management function to ensure the quality, integrity, security, and usability of the data collected by an organization."

Data Governance: The definitive Guide

Data Quality

High Quality Data

- Accurate
- Complete
- Discoverable
- Available in a timely manner

Exactly what stakeholders expect

Low Quality Data

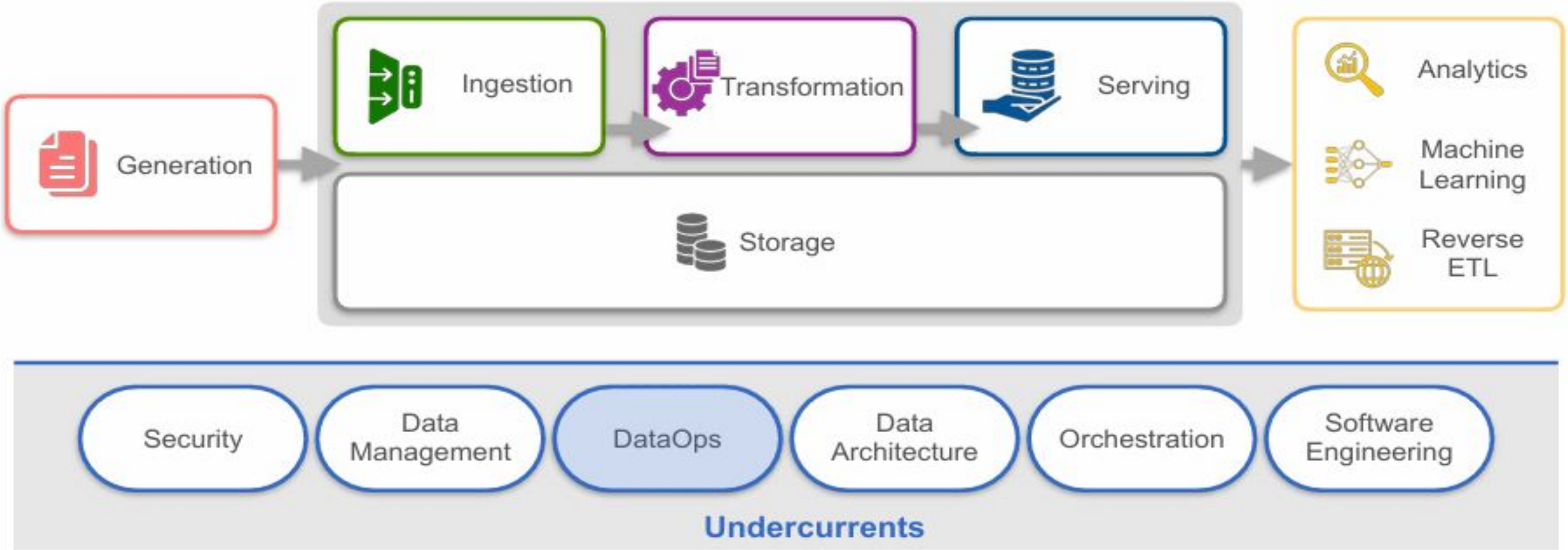
- Inaccurate
- Incomplete
- Hard to find
- Late

Unusable

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Dataops

The Data Engineering Lifecycle & Undercurrents



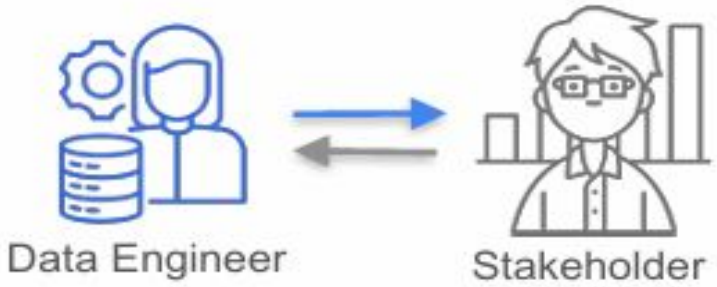
DataOps

DataOps

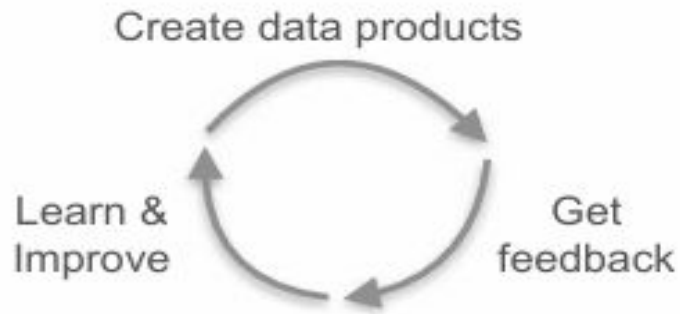
Improves the development process and quality of data products.

It's a set of cultural habits and practices.

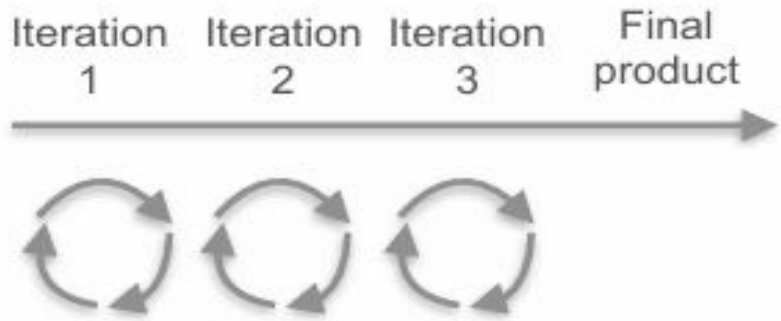
Communication & Collaboration



Continuous Improvement

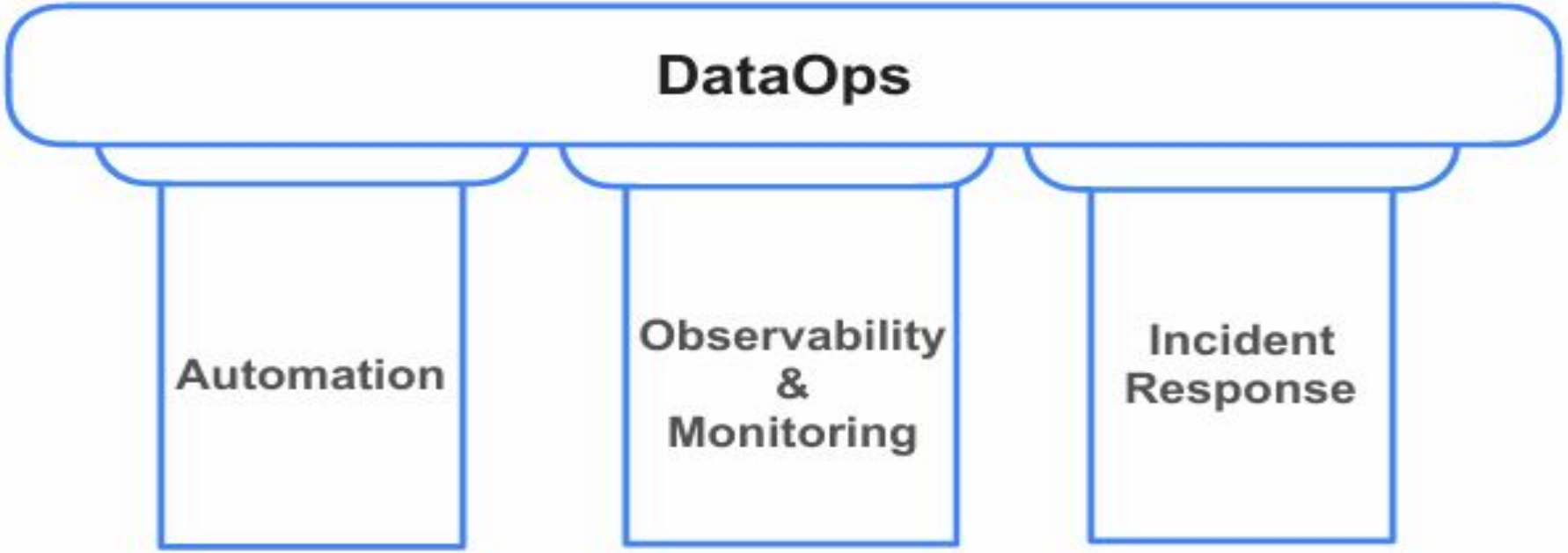


Rapid Iteration



DevOps practices ⟵ **Agile methodology**

Pillars of DataOps



Goal: Provide high-quality data products

Pillar 1: Automation

DevOps
(Applies to software build)

Continuous Integration and Continuous Delivery (CI/CD)

CI/CD automation results in:

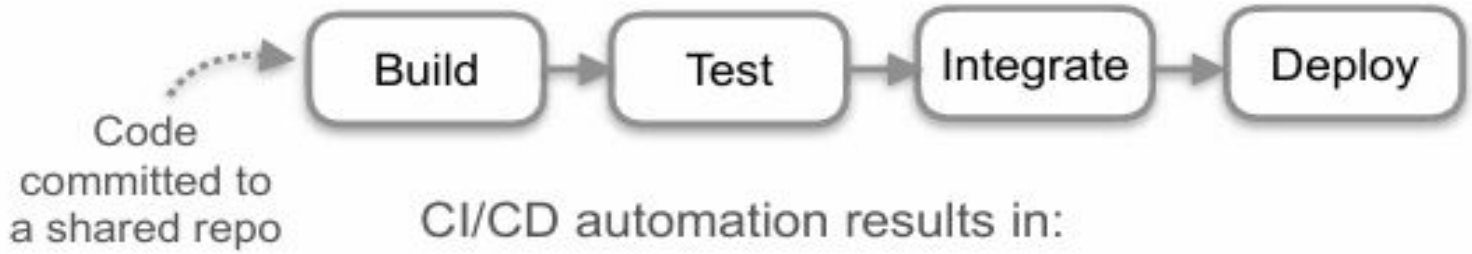
- faster deployment
- fewer errors

DataOps

Pillar 1: Automation

DevOps
(Applies to software build)

Continuous Integration and Continuous Delivery (CI/CD)



CI/CD automation results in:

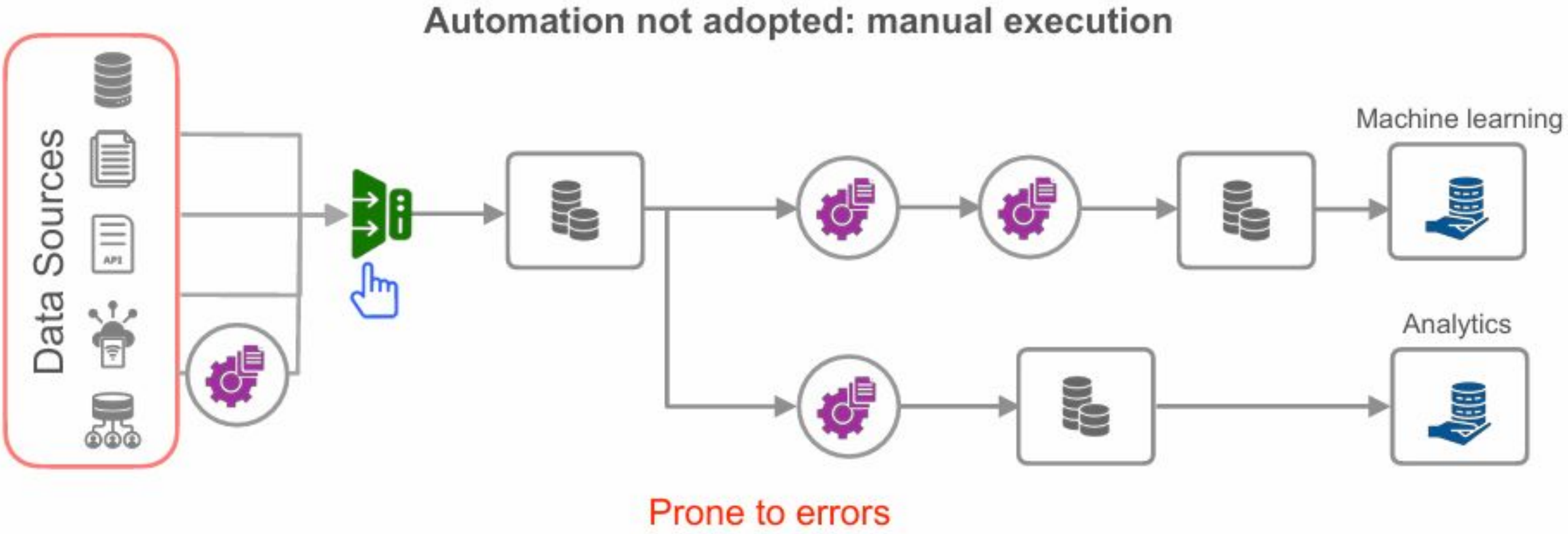
- faster deployment
- fewer errors

DataOps
(Applies to data processing)

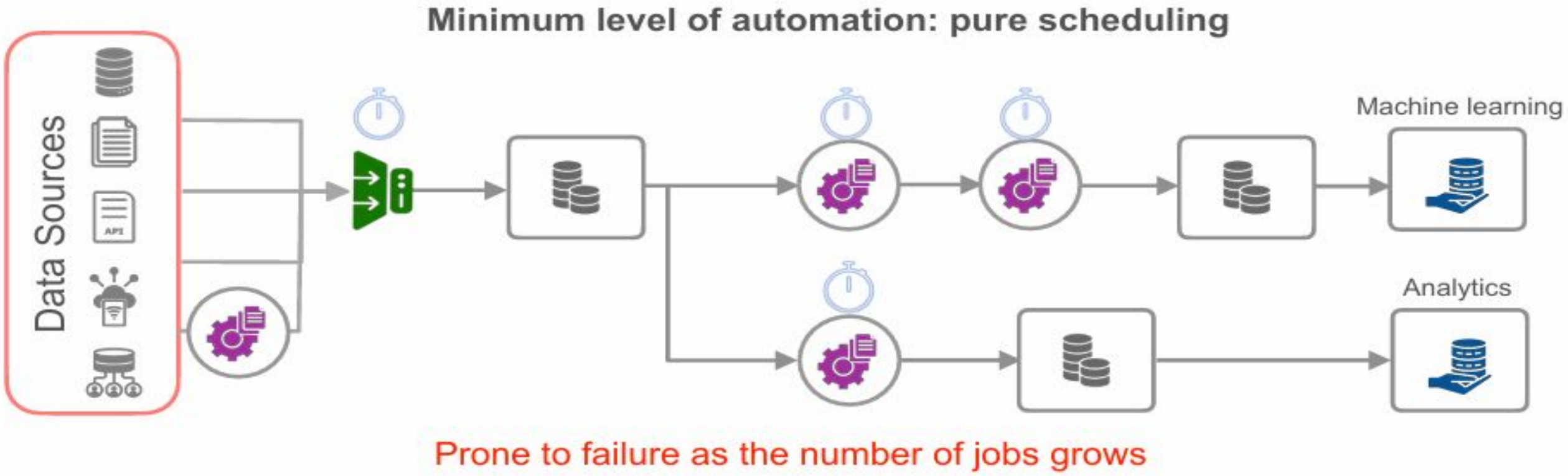
Automated change management:

- Code
- Configuration
- Environment
- Data processing pipelines
- Data

Pillar 1: Automation

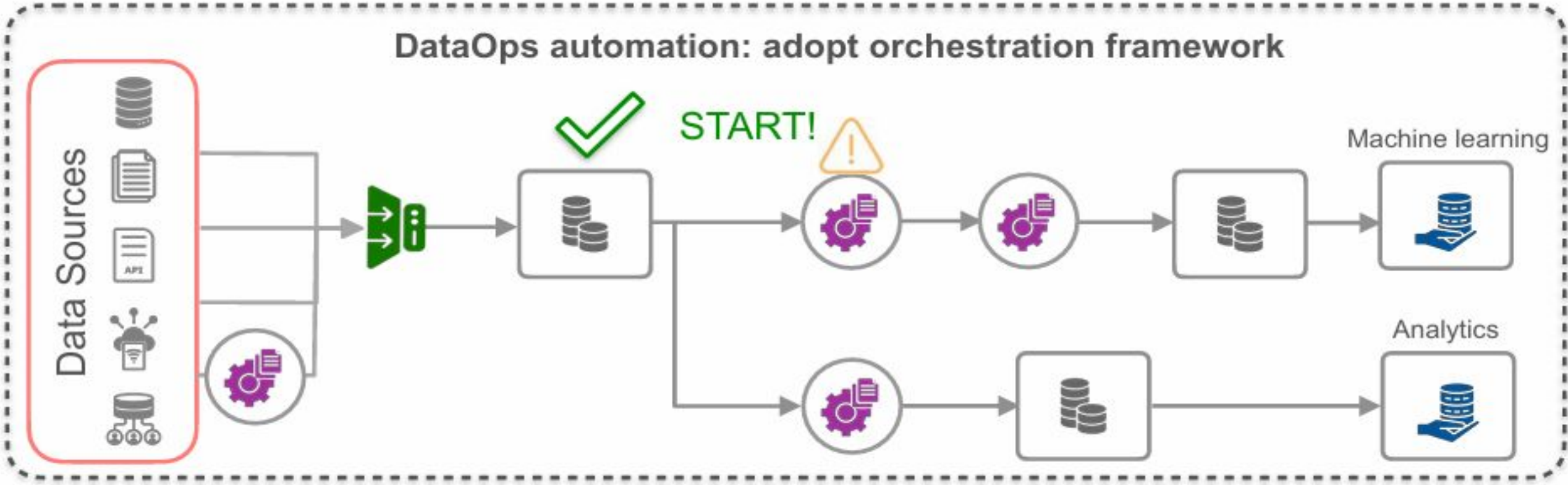


Pillar 1: Automation



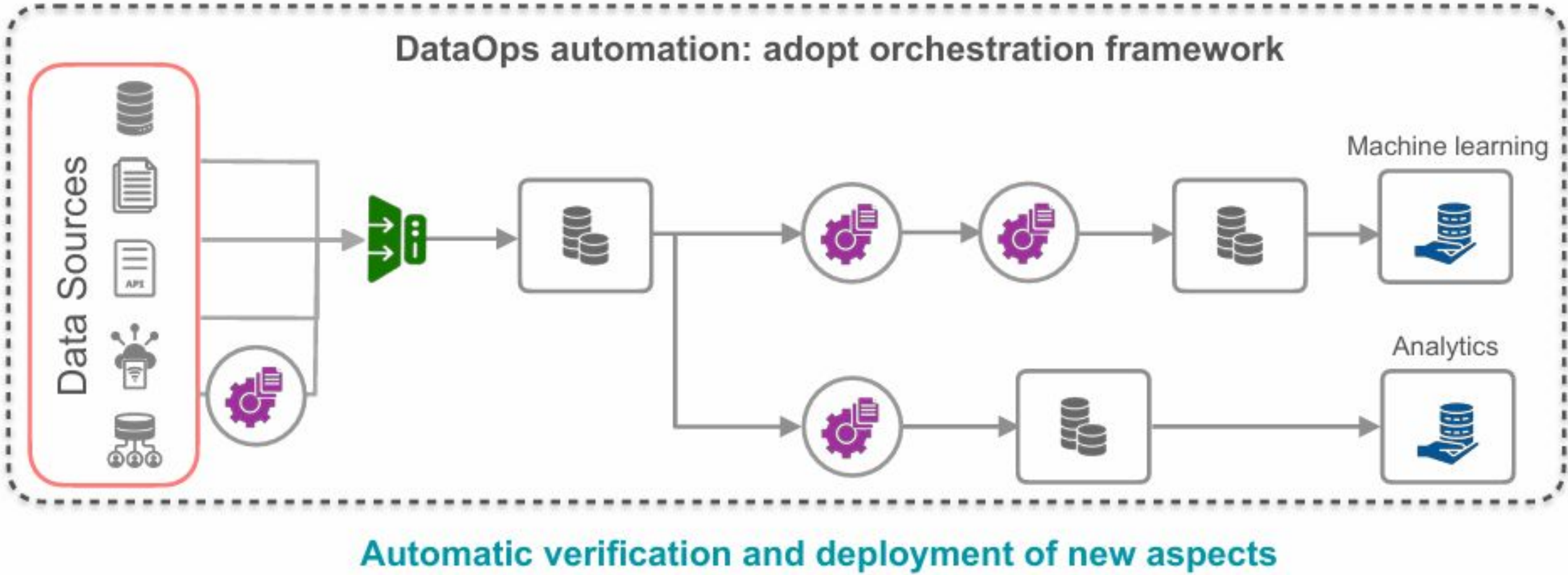
Moves from manual execution → scheduled automation → orchestration using tools like Apache Airflow.Orchestration manages task dependencies and error handling automatically, enabling robust, scalable pipelines.

Pillar 1: Automation

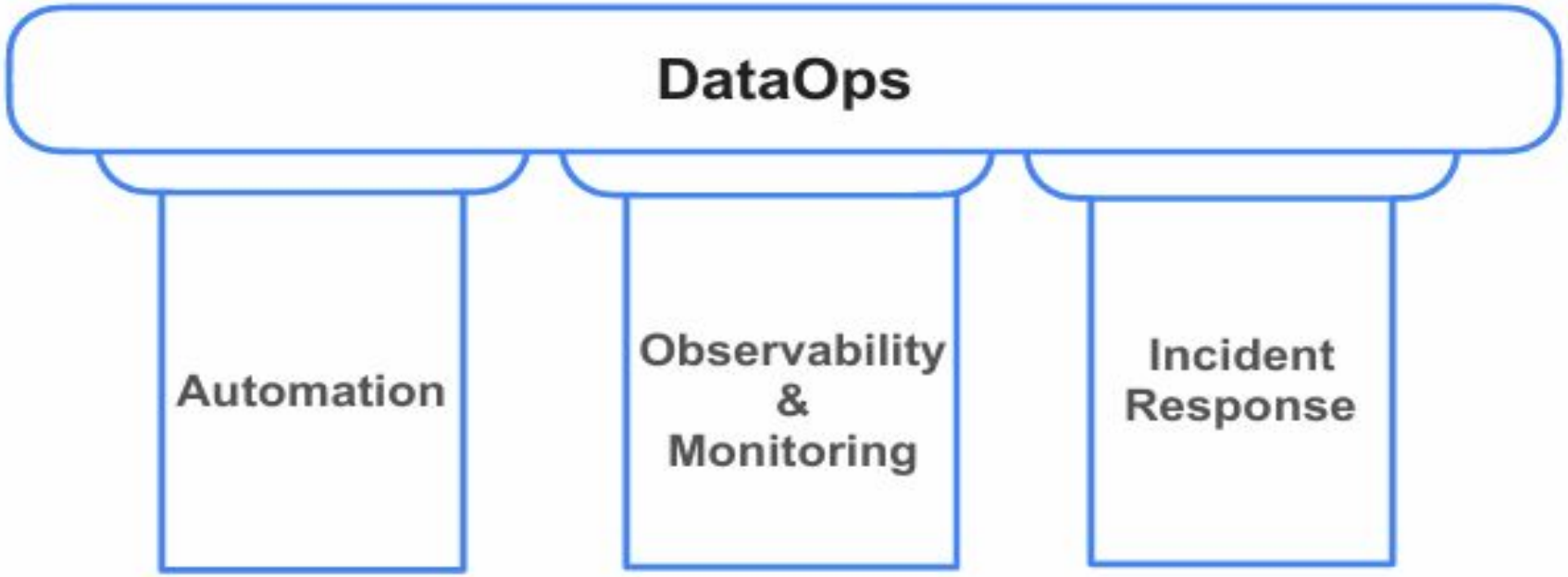


Checks the dependencies between tasks before each task is run

Pillar 1: Automation



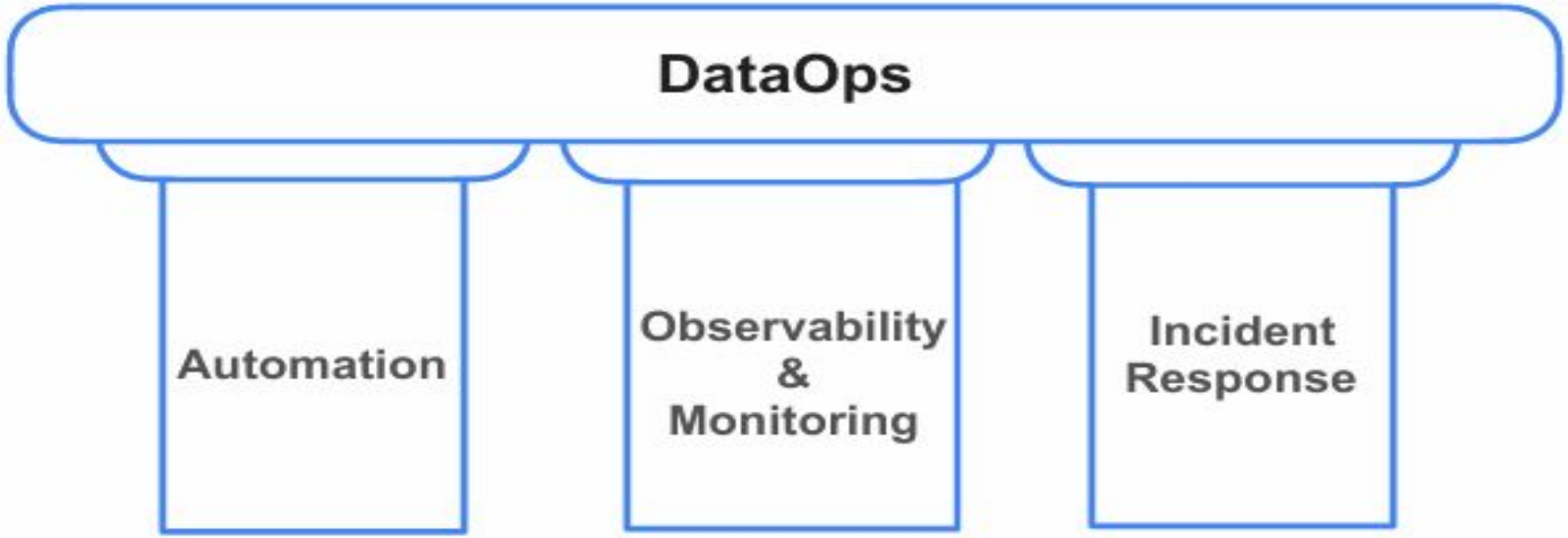
Pillar 2: Observability & Monitoring



"Everything fails all the time"

- Werner Vogels

Pillar 3: Incident Response

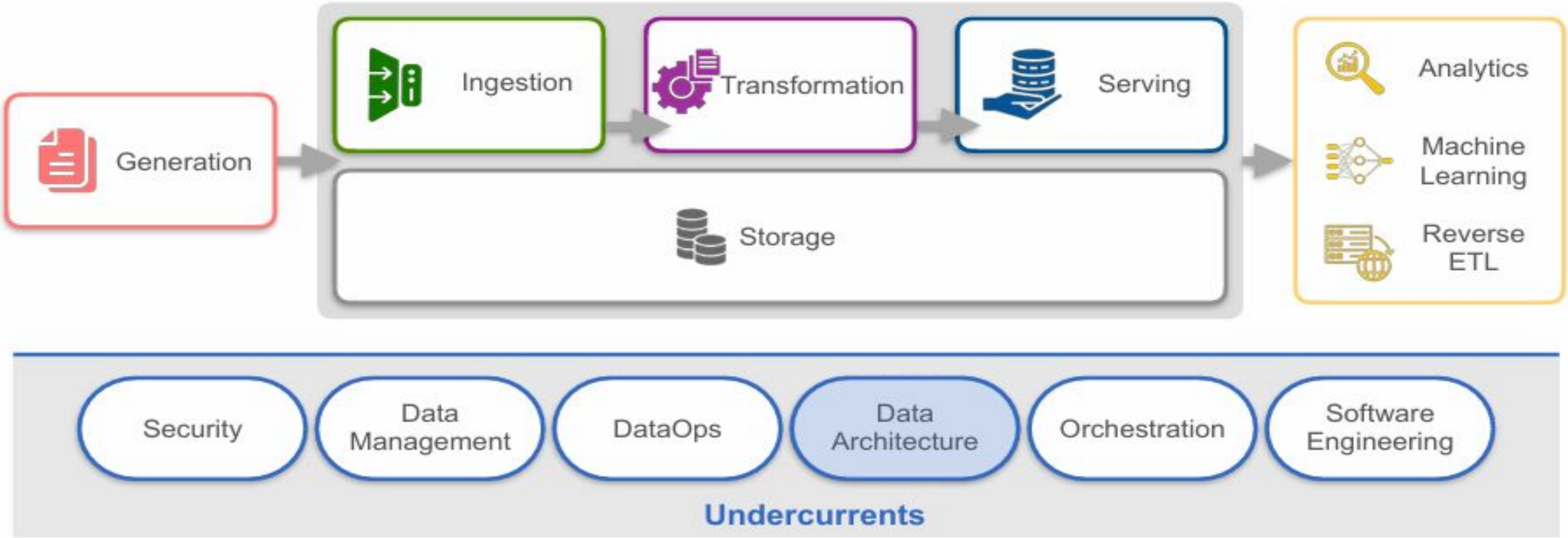


- ✓ Rapidly identify the incident's root causes
- ✓ Quickly resolve an incident
- ✓ Identify technology and tools
- ✓ Coordinate the efforts of the data team

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

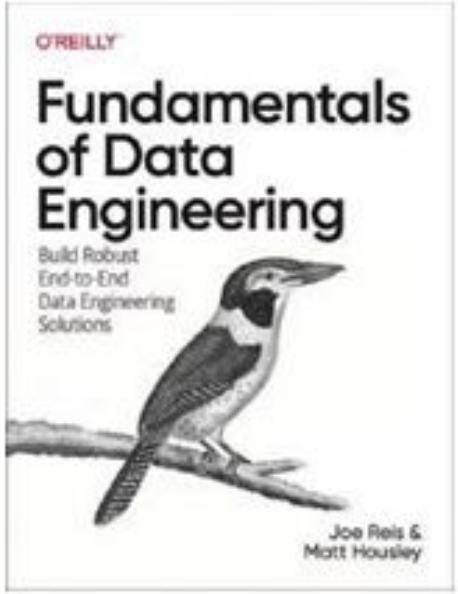
Data architecture

The Data Engineering Lifecycle & Undercurrents



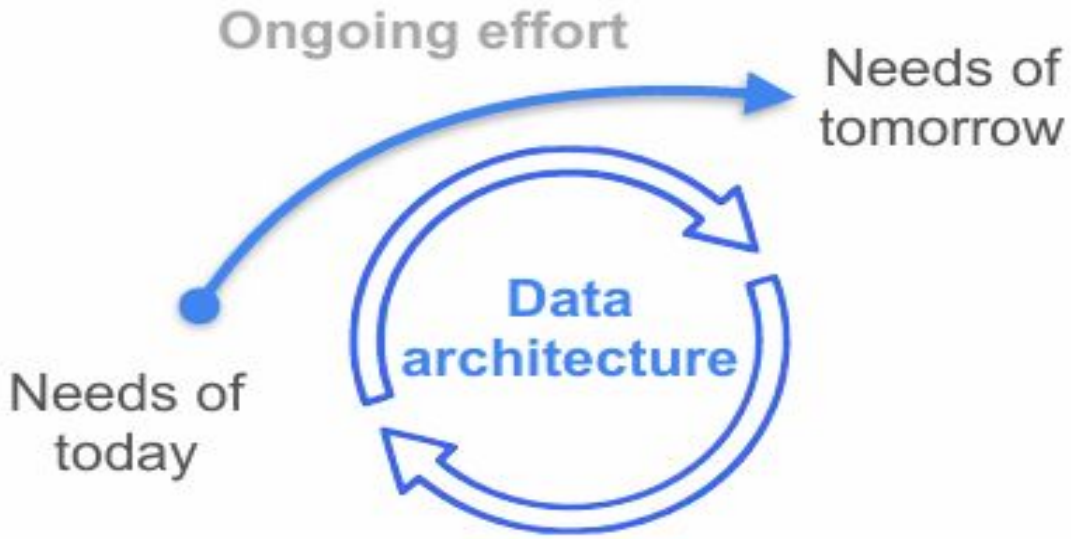
Data Architecture

"Data architecture is the design of systems to support the evolving data needs of an enterprise, achieved by flexible and reversible decisions reached through a careful evaluation of trade-offs"



Data Architecture

"Data architecture is the design of systems to support the evolving data needs of an enterprise, achieved by flexible and reversible decisions reached through a careful evaluation of trade-offs"



Data Architecture

"Data architecture is the design of systems to support the evolving data needs of an enterprise, achieved by flexible and reversible decisions reached through a careful evaluation of trade-offs"

Data Architecture

"Data architecture is the design of systems to support the evolving data needs of an enterprise, achieved by flexible and reversible decisions reached through a careful evaluation of trade-offs"

Trade-offs

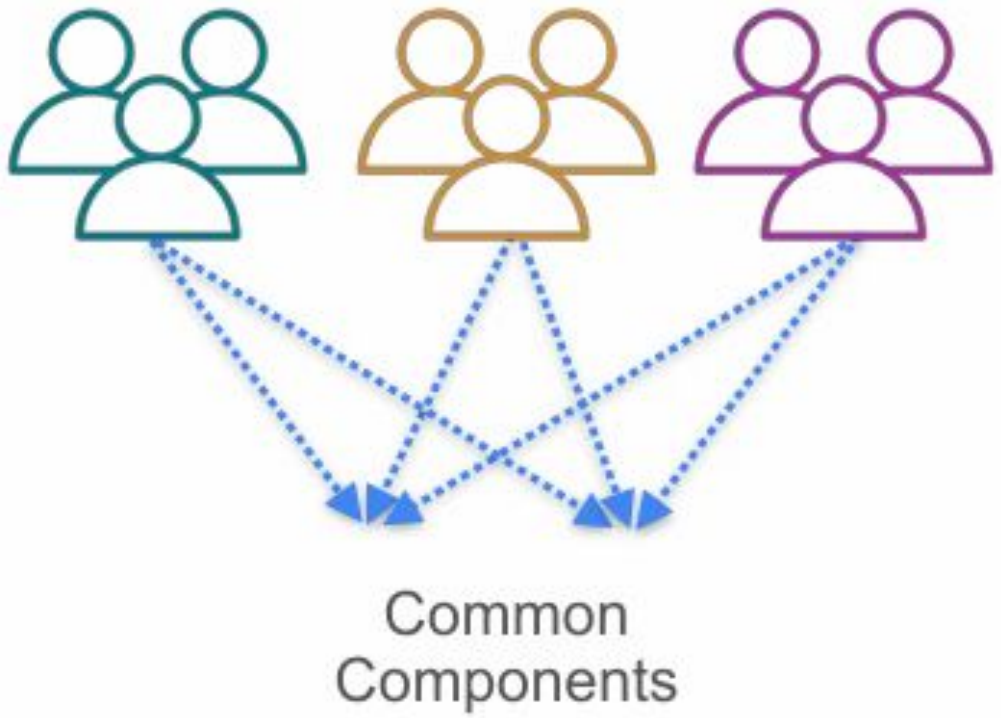
- Performance
- Cost
- Scalability
- …

Data Architecture

Select shared tools that support both individual project needs and cross-team collaboration.

Principle of Good Data Ar ecture

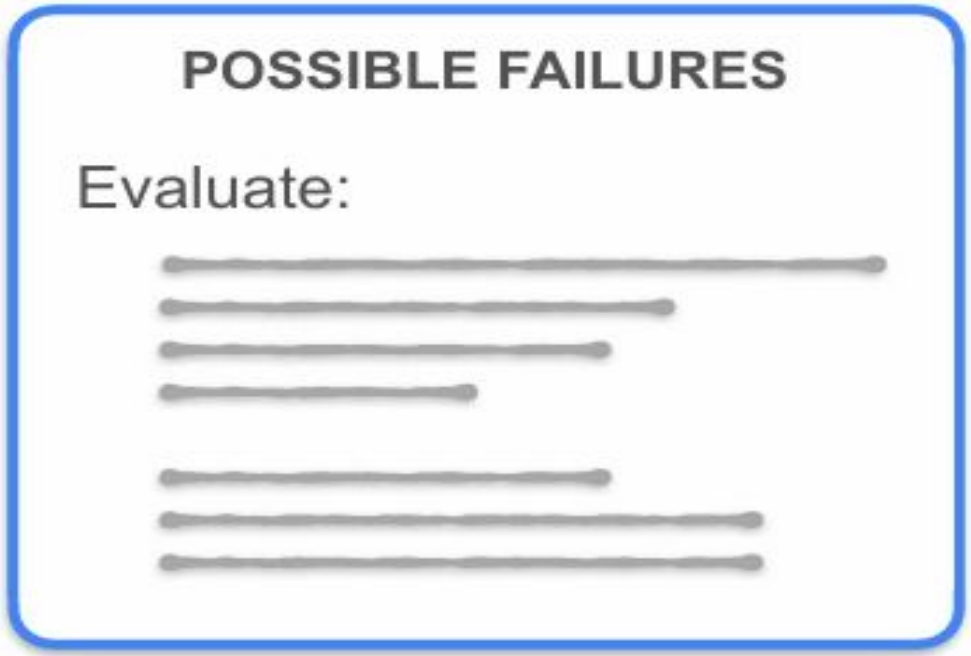
1. Choose common components wisely



Systems must handle both expected operations and unexpected breakdowns.

Principle of Good Data Architecture

1. Choose common components wisely
2. Plan for failure!



Build systems that can scale up with demand and scale down to save costs.

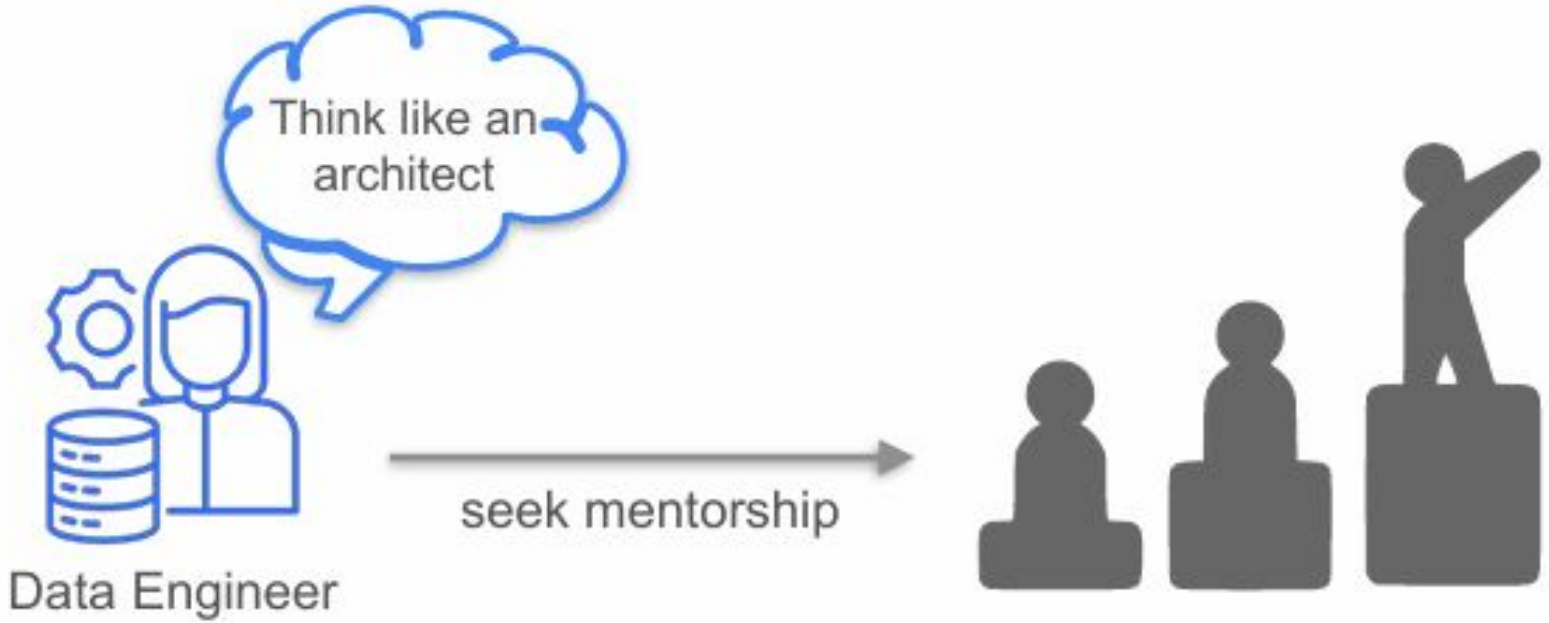
Principle od Data Architecture

1. Choose con components wisely
2. Plan for failu
3. Architect for scalability

Developing architectural thinking prepares you to lead and mentor as your career progresses.

Principle d Data Architecture

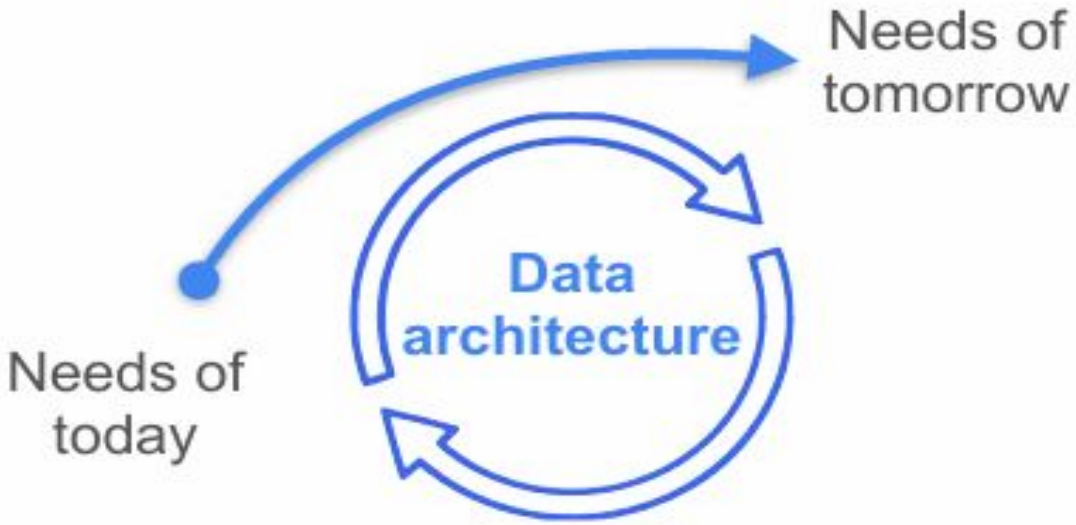
1. Choose co components wisely
2. Plan for failu
3. Architect for s alability
4. Architecture is leadership



Architecture is an ongoing process, constantly evolving with organizational needs.

Principle od Data Architecture

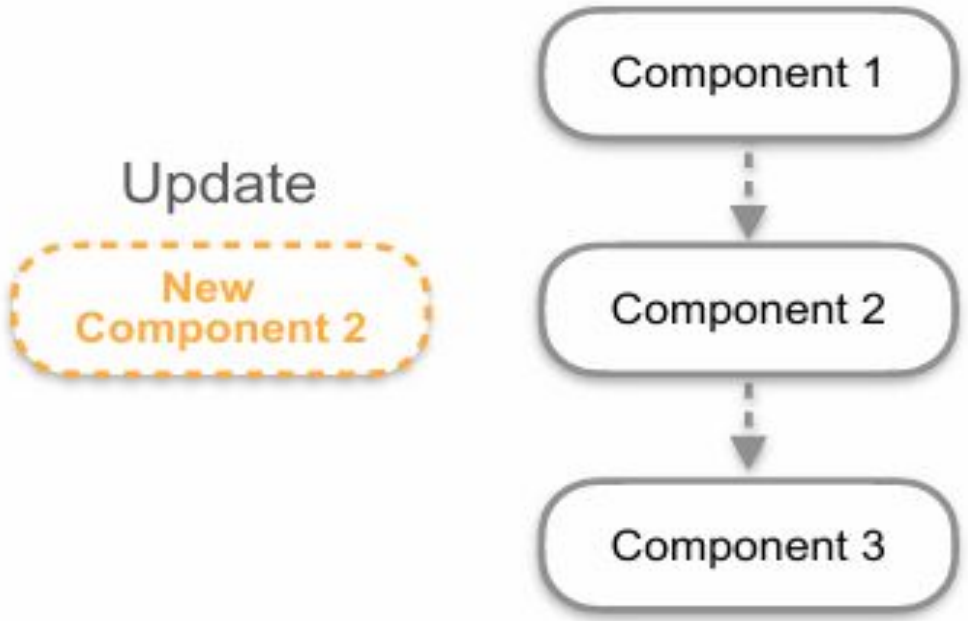
1. Choose co components wisely
2. Plan for fail
3. Architect for alability
4. Architecture leadership
5. Always be architecting



Design components that can be independently replaced or upgraded.

Principle d Data Architecture

1. Choose co mponents wisely
2. Plan for failu
3. Architect for bility
4. Architecture is adership
5. Always be arc tecting
6. Build loosely coupled systems
7. Make reversible decisions



Ensure design choices can be changed without overhauling the entire system.

Principle od Data Architecture

1. Choose c components wisely
2. Plan for fai
3. Architect fo ability
4. Architecture eadership
5. Always be a itecting
6. Build loosely coupled systems
7. Make reversible decisions
8. Prioritize security

Principle of least privilege

Zero-trust principle

Security is fundamental and must be integrated from the start. Design cloud-based systems that are cost-effective and financially transparent by aligning finance with DevOps/DataOps principles.

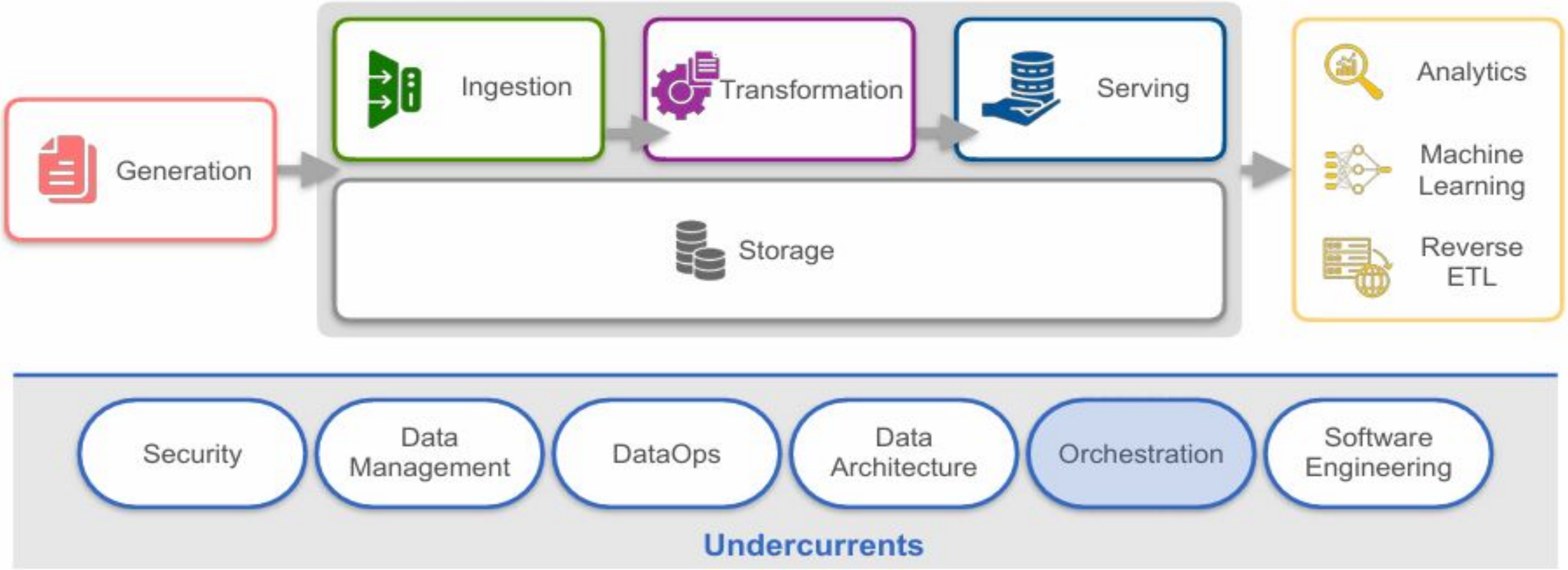
Principle Data Architecture

1. Choose omponents wisely
2. Plan fo
3. Archit calability
4. Arch e is leadership
5. Alw be architecting
6. B loosely coupled systems
7. M ke reversible decisions
8. Prioritize security
9. Embrace FinOps

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

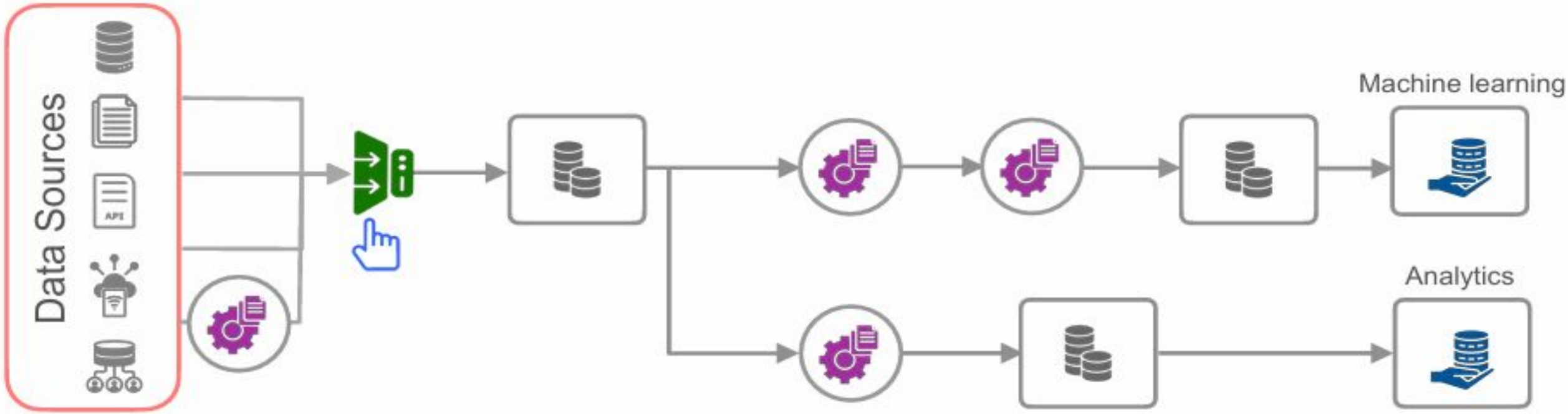
Orchestration

The Data Engineering Lifecycle & Undercurrents



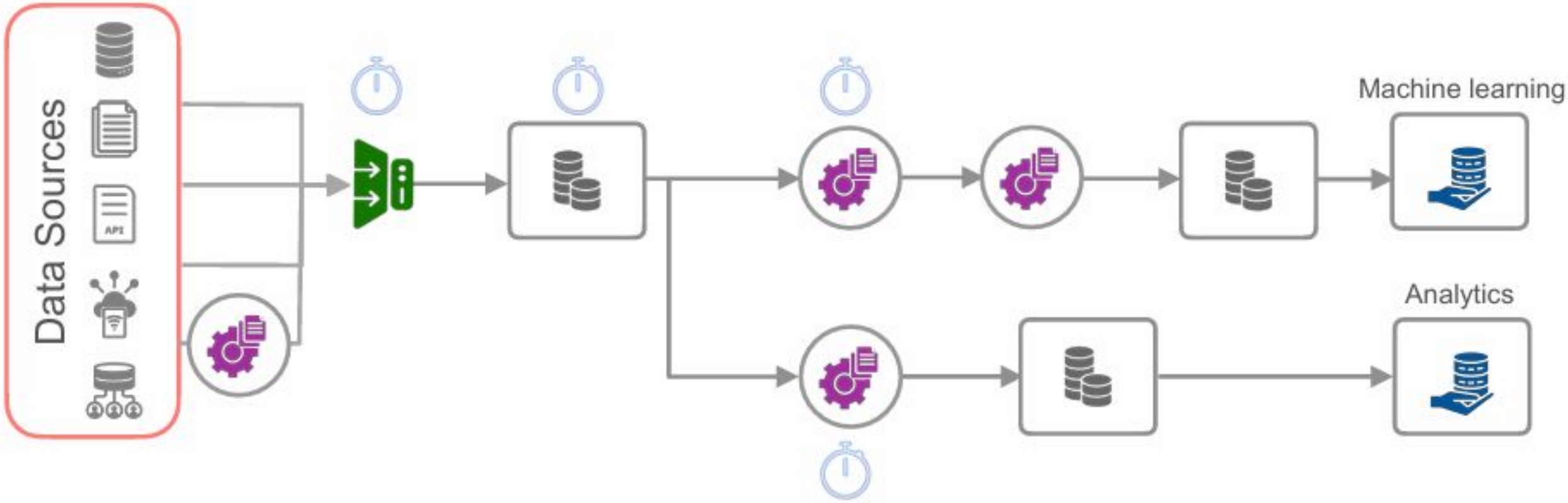
data engineer coordinates and manages tasks within a data pipeline to ensure smooth, accurate data flow. Initially, manual execution of tasks may be useful during prototyping, but it becomes unsustainable for long-term data processing.

Manual Execution

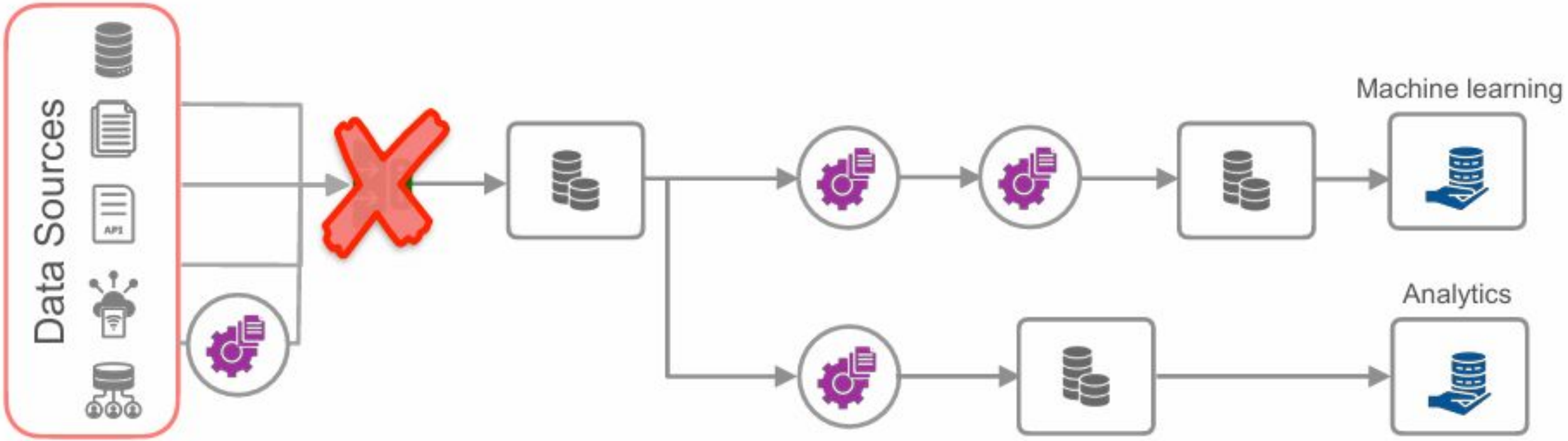


Tasks are scheduled to run at fixed times but can lead to failures if earlier tasks fail or take longer than expected.

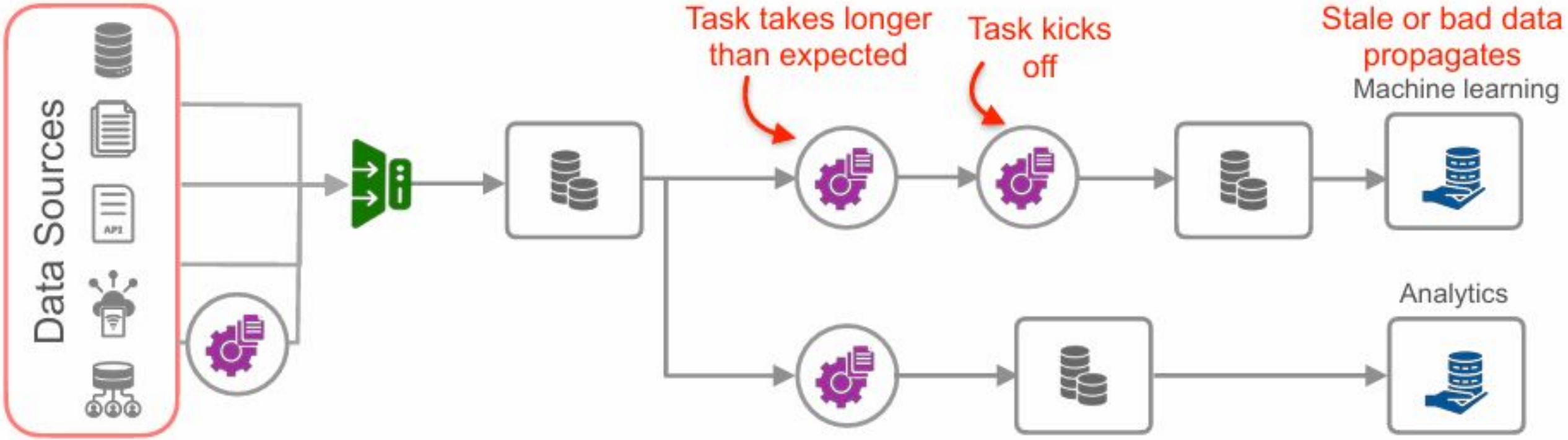
Pure Scheduling



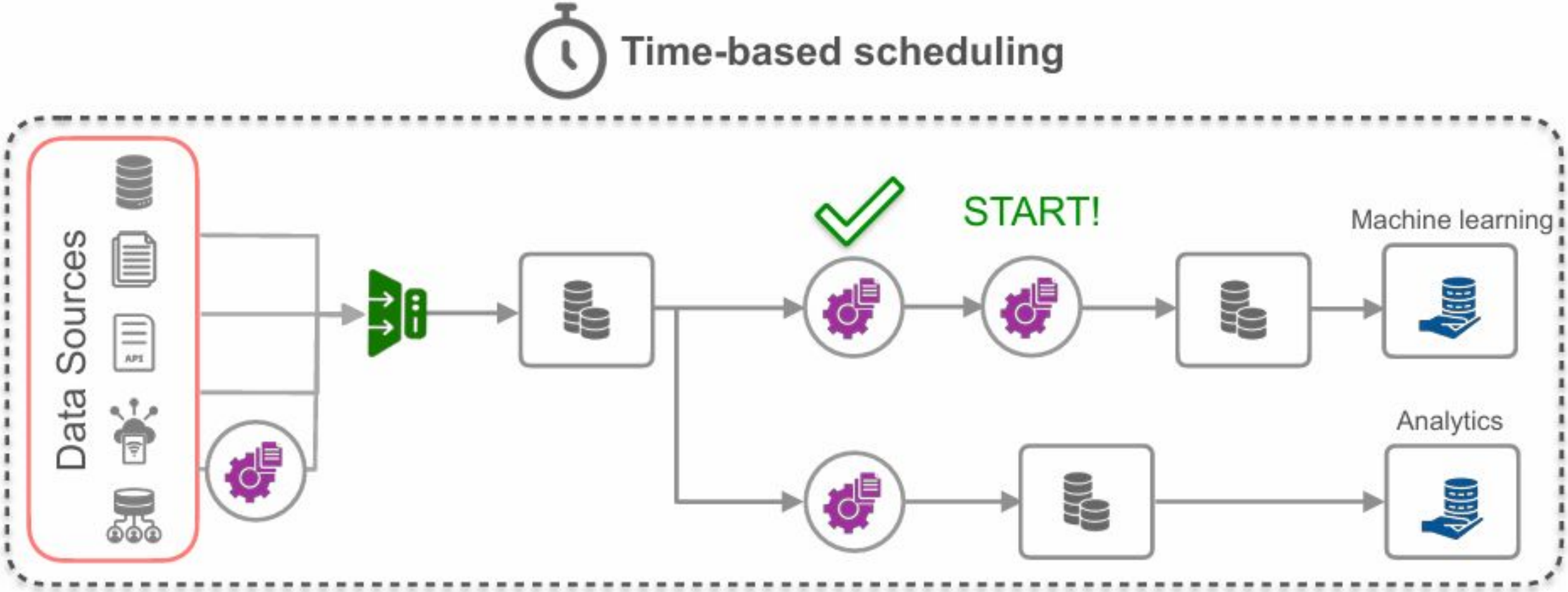
Pure Scheduling



Pure Scheduling



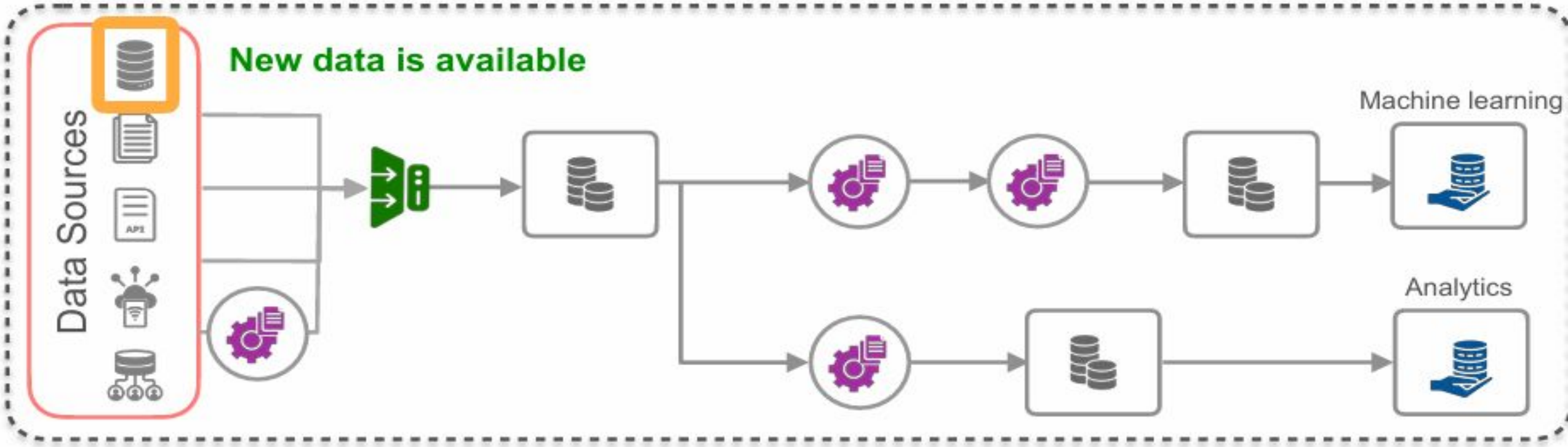
Orchestration Frameworks



Orchestration frameworks:

- Automate pipeline with complex dependencies
- Monitor pipeline

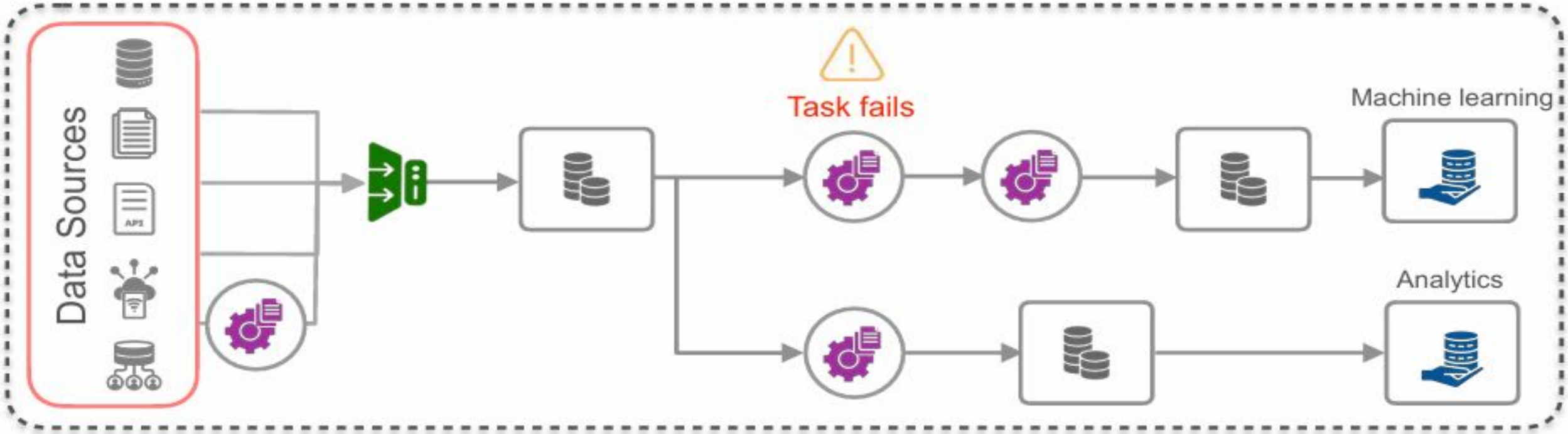
Event-based triggers



Orchestration frameworks:

- Automate pipeline with complex dependencies
- Monitor pipeline

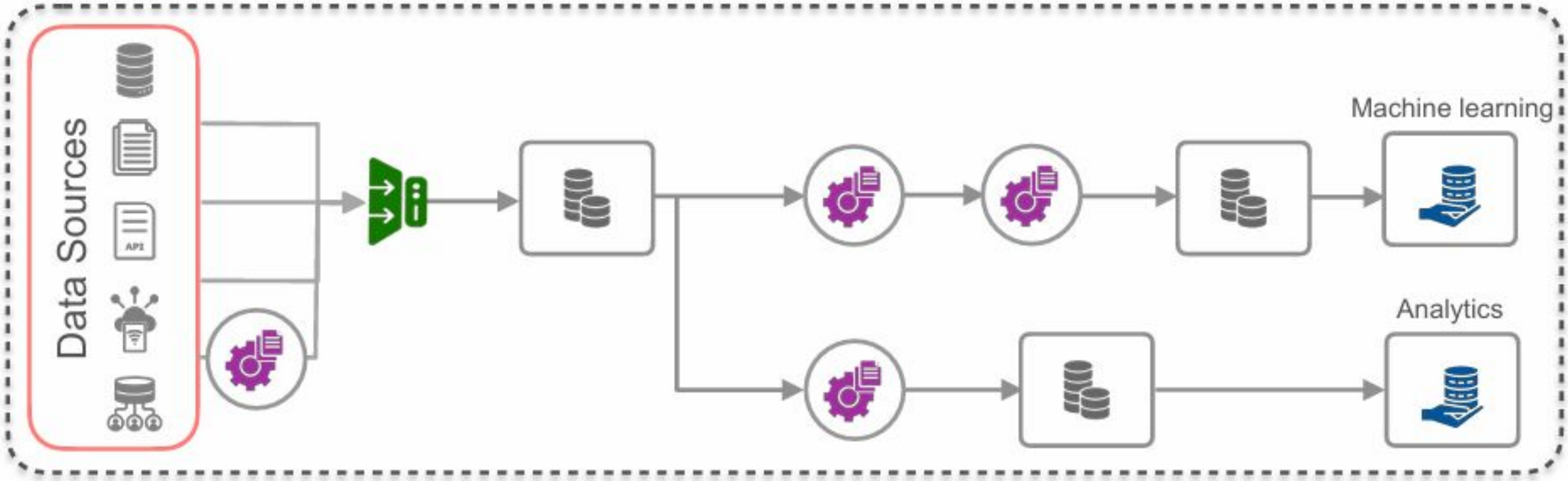
Set up monitoring & alerts



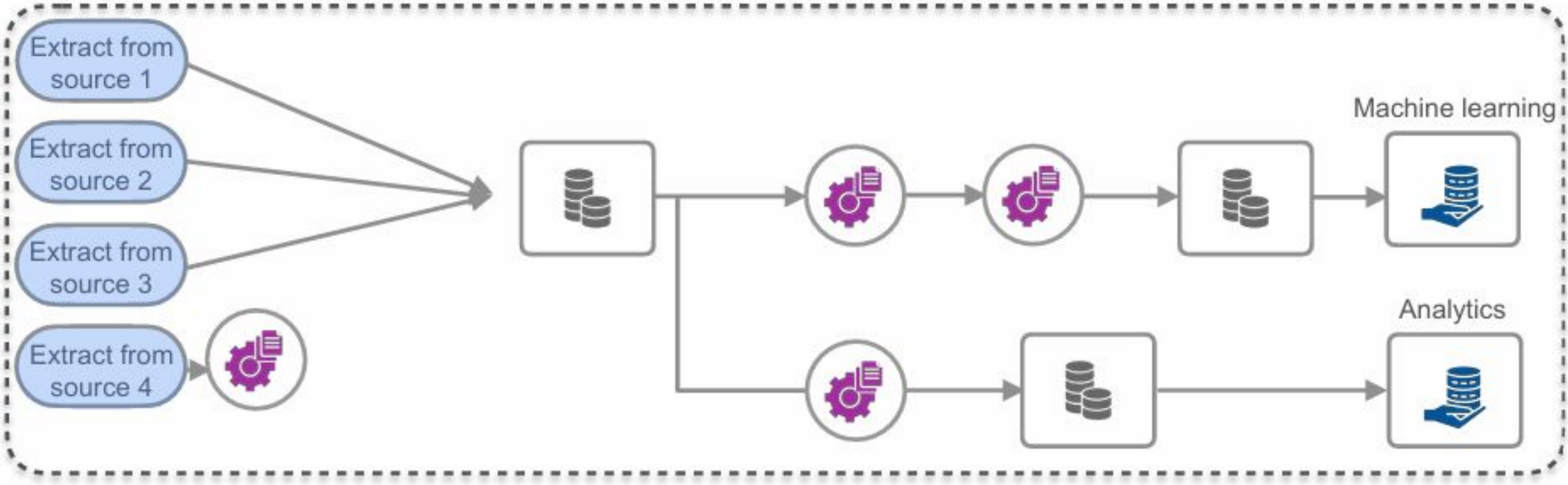
Orchestration frameworks:

- Automate pipeline with complex dependencies
- Monitor pipeline

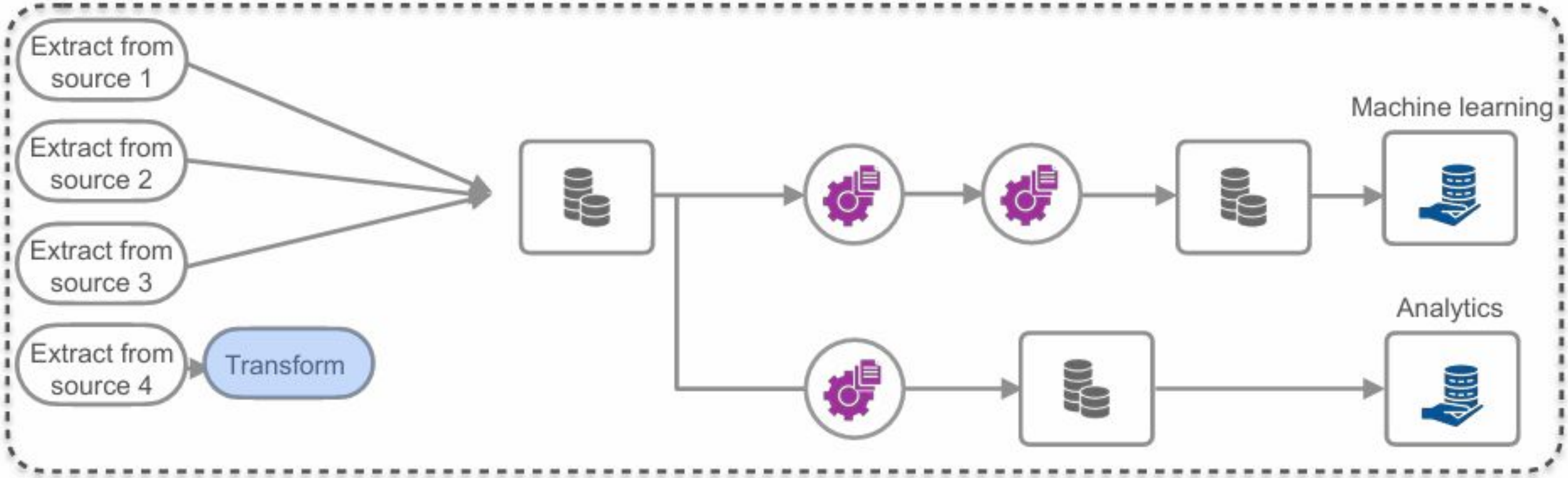
Directed Acyclic Graph



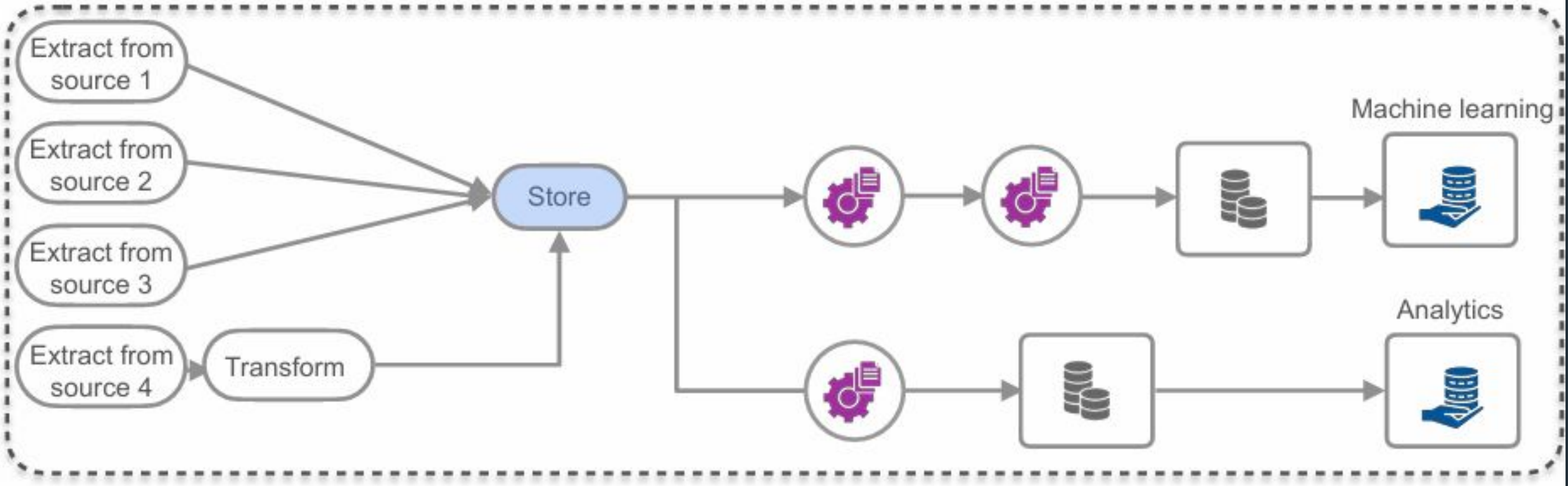
Directed Acyclic Graph (DAG)



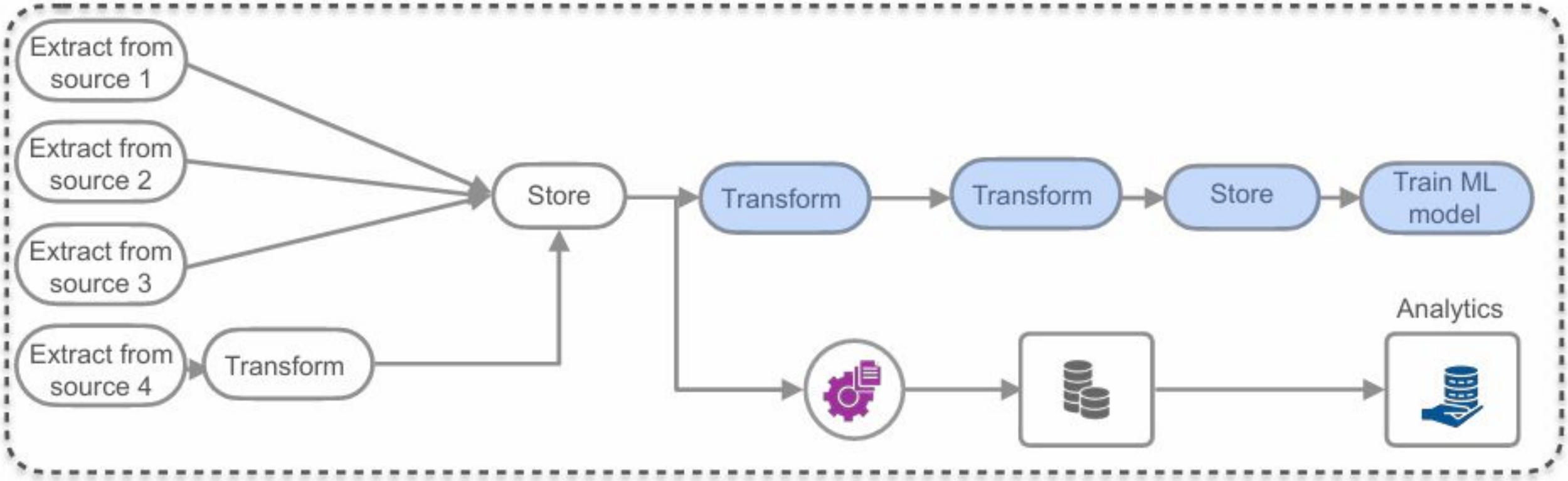
Directed Acyclic Graph (DAG)



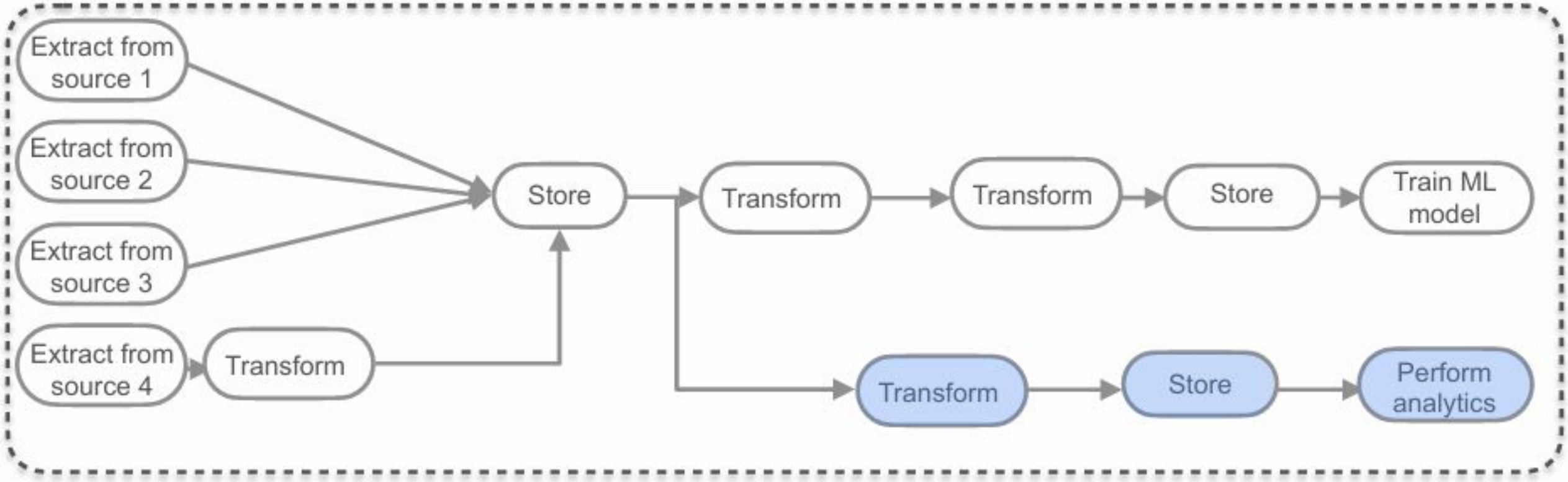
Directed Acyclic Graph (DAG)



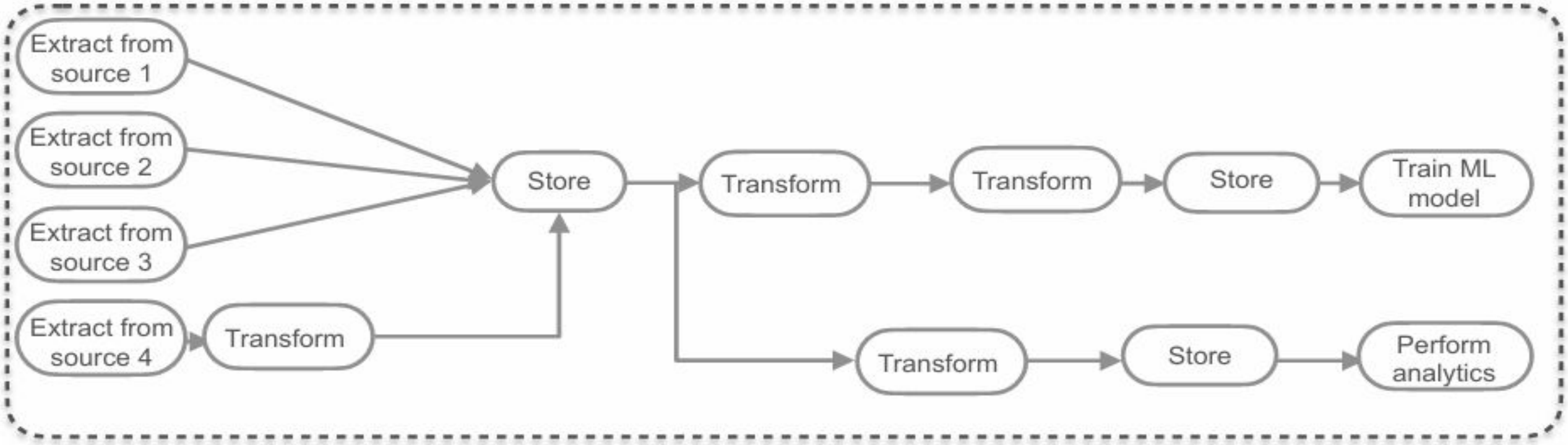
Directed Acyclic Graph (DAG)



Directed Acyclic Graph (DAG)

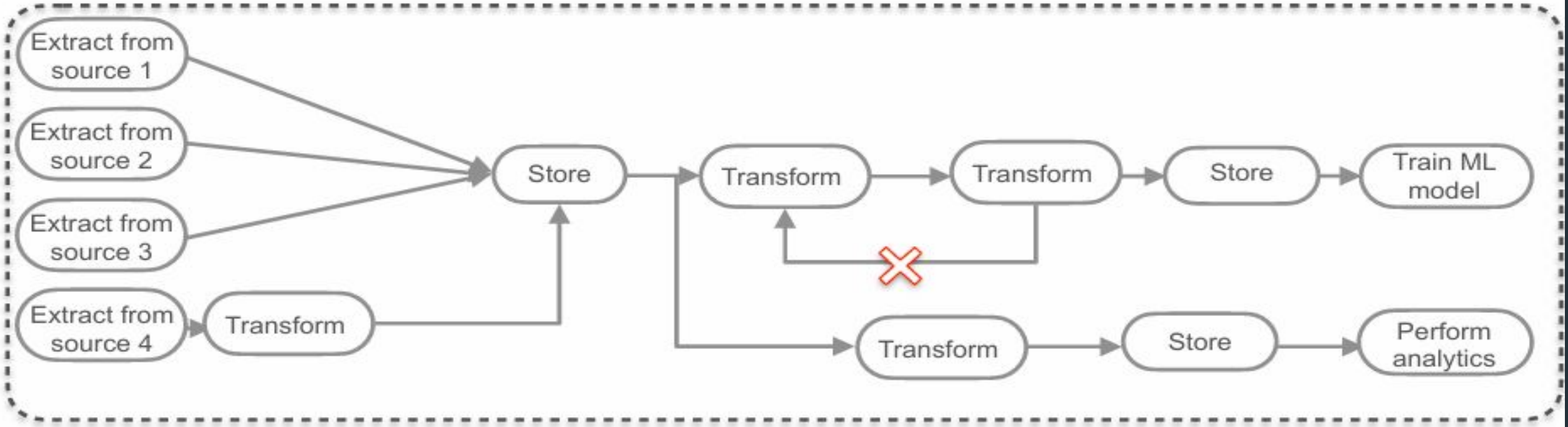


Directed Acyclic Graph (DAG)



Directed Data flows in one direction

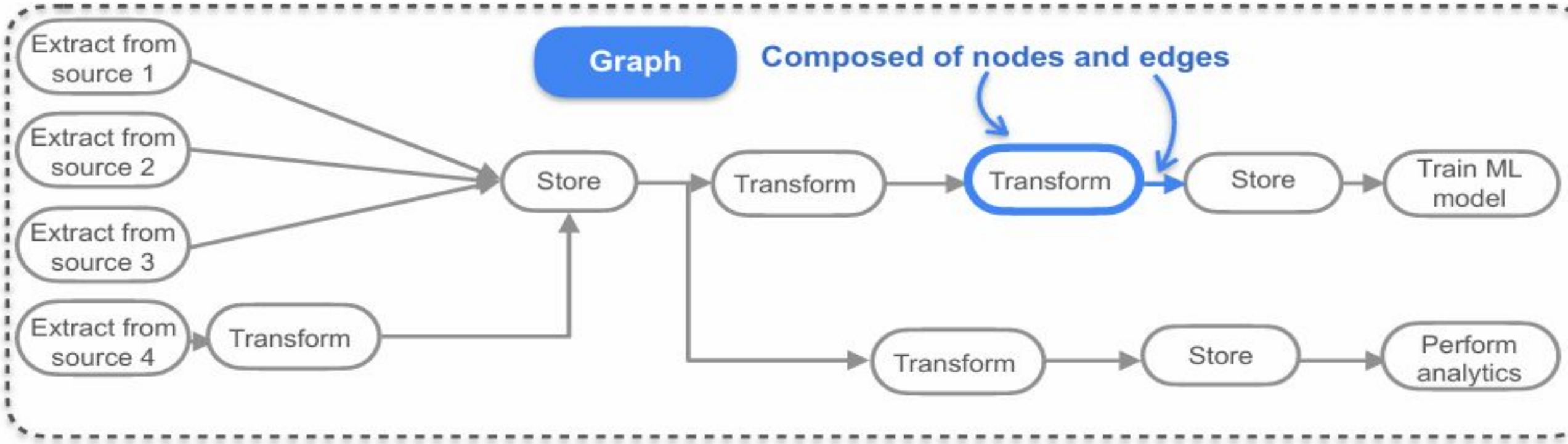
Directed Acyclic Graph (DAG)



Directed Data flows in one direction

Acyclic Data doesn't flow backward

Directed Acyclic Graph (DAG)



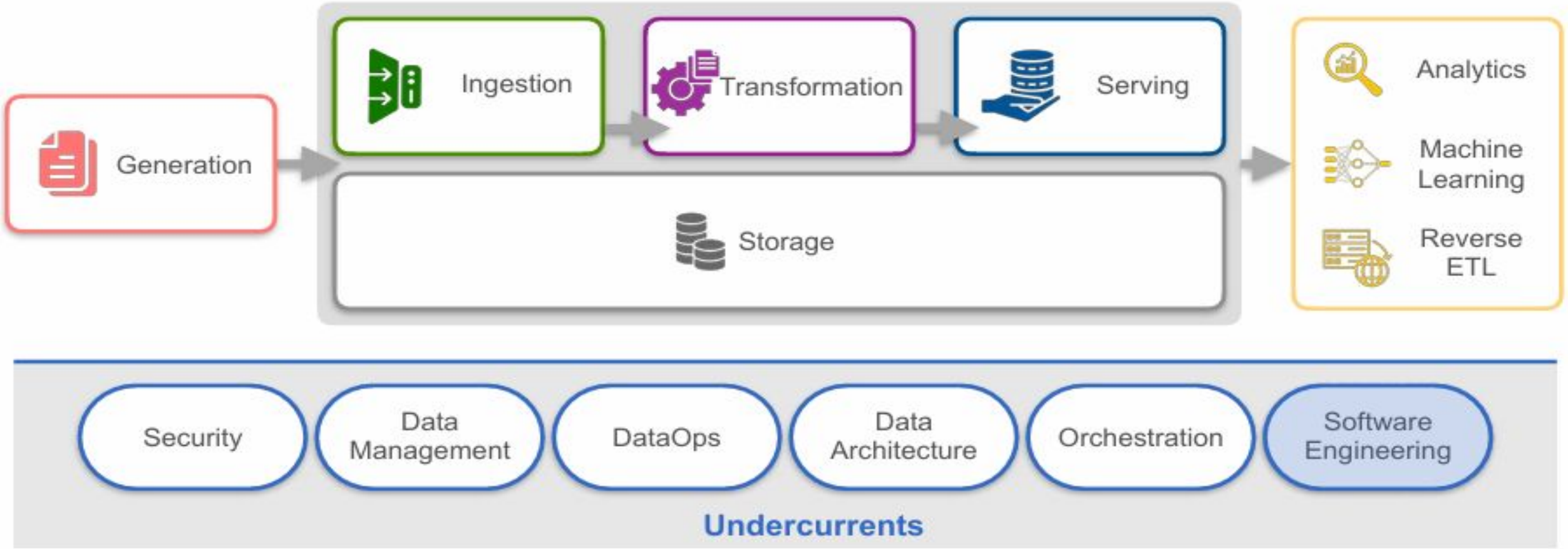
Directed Data flows in one direction

Acyclic Data doesn't flow backward

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Software Engineering

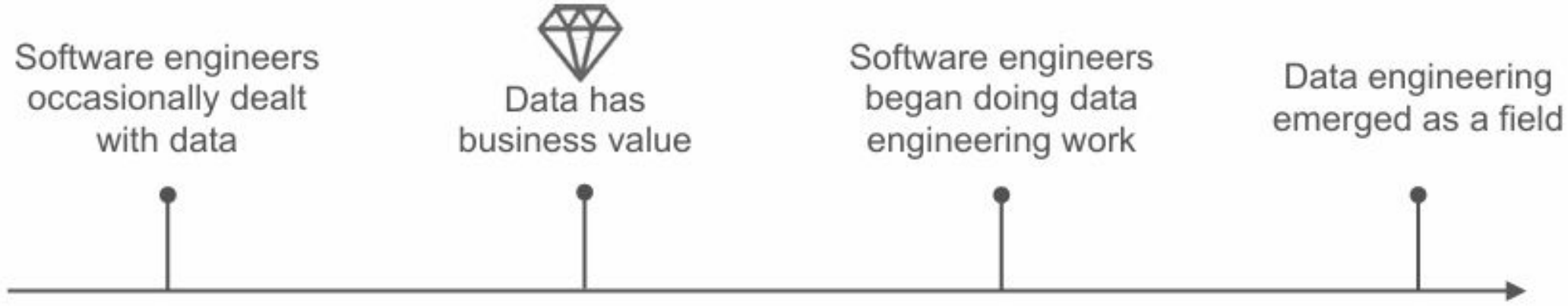
The Data Engineering Lifecycle & Undercurrents



Software Engineering

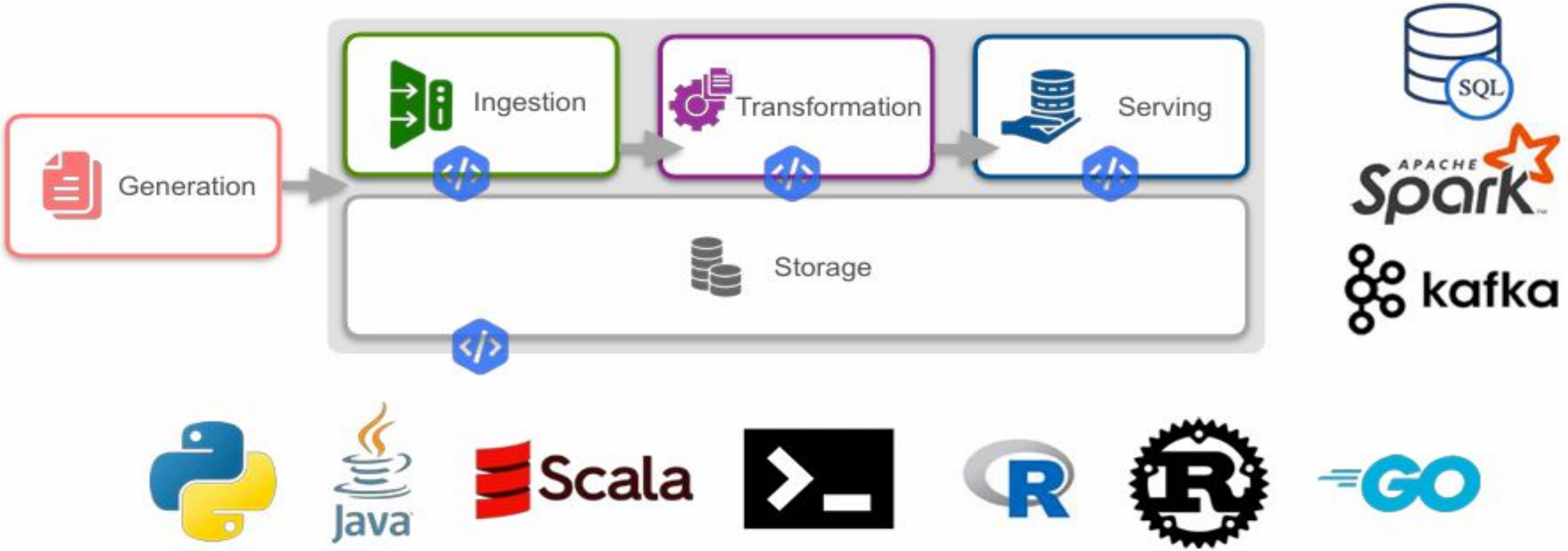
Software engineering

The design, development, deployment, and maintenance of software applications.

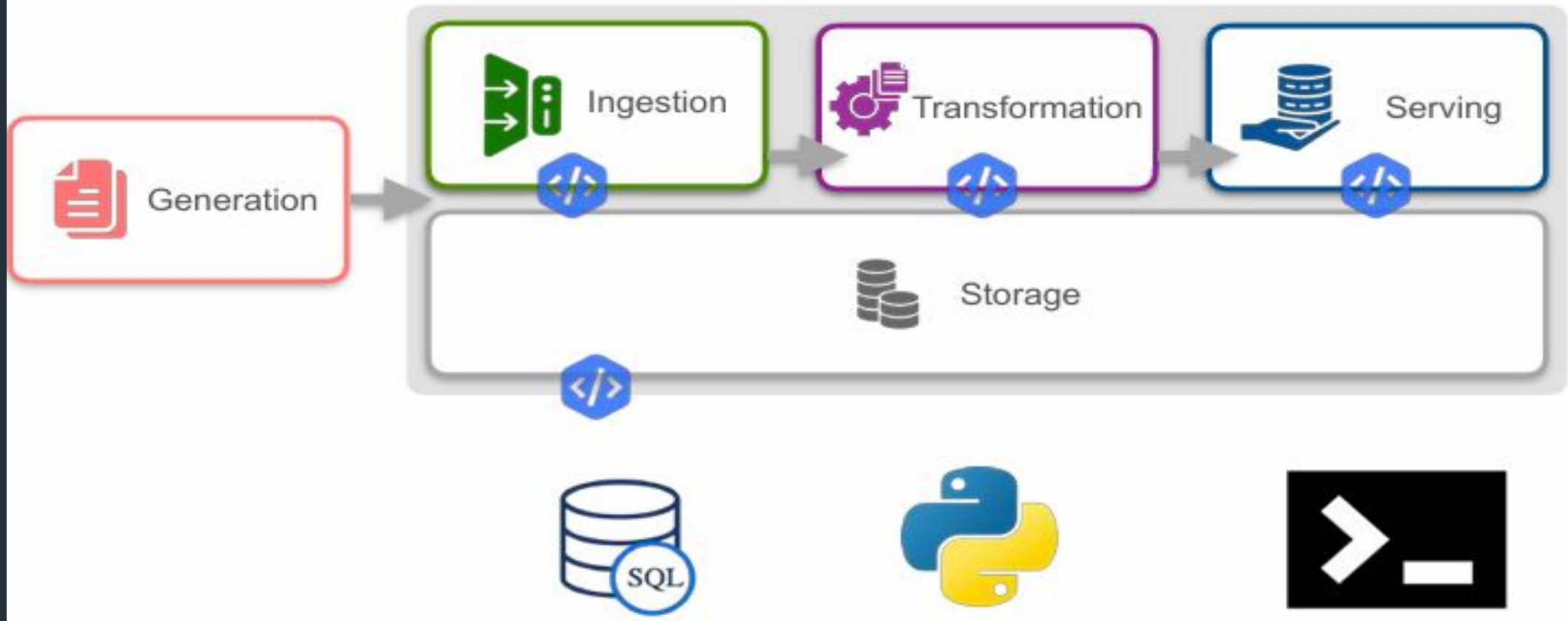


Core coding tasks in data engineering include:Data ingestion, transformation, and serving.Writing code using languages and frameworks like SQL, Python, Bash, Spark, and Kafka.

Writing Code as a Data Engineer



Writing Code as a Data Engineer



Writing Code as a Data Engineer

Other coding use cases:

- Open source frameworks
- Infrastructure as code
- Pipeline as code
- Everyday general-purpose problem solving