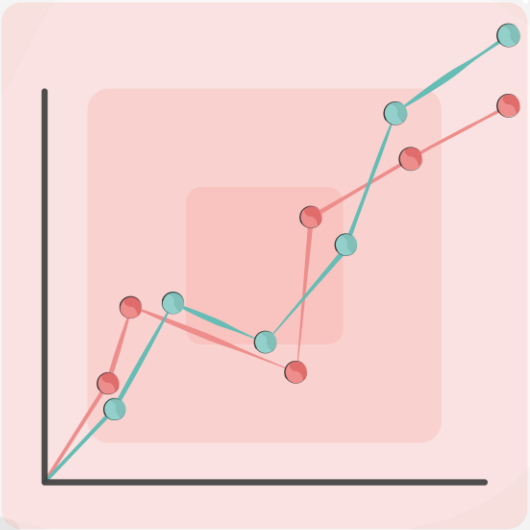
Event-driven architectures fundamentals

Welcome to today's webinar.



Did you know that 85% of organisations use event-driven architecture, and 94% of them plan to expand its use?

Real-world case study

Tesco’s management of product inventory

- **Background:** Tesco faced inventory management challenges with traditional batch processing.
- **Challenge:** Needed real-time inventory visibility and optimised supply chain.
- **Solution:** Implemented Apache Kafka for real-time data processing.

Tesco: *Using data analysis to make better business decisions*

Benefits:

- Real-time inventory updates
- Improved customer experience
- Optimised supply chain
- Scalable solution

Knowledge check poll

Imagine you are working for an e-commerce company that wants to improve its real-time inventory management system.

Which of the following best describes how an event-driven architecture can help achieve this goal?

A. By using batch processing to update inventory levels at the end of each day

B. By implementing a request-response model to check inventory levels before each purchase

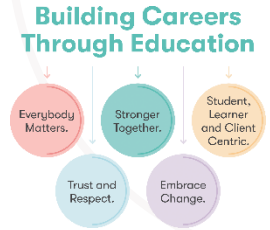
C. By using an event-driven architecture to update inventory levels in real-time as purchases are made

Submit your responses to the chat!

Feedback: C – By using an event-driven architecture to update inventory levels in real-time as purchases are made.

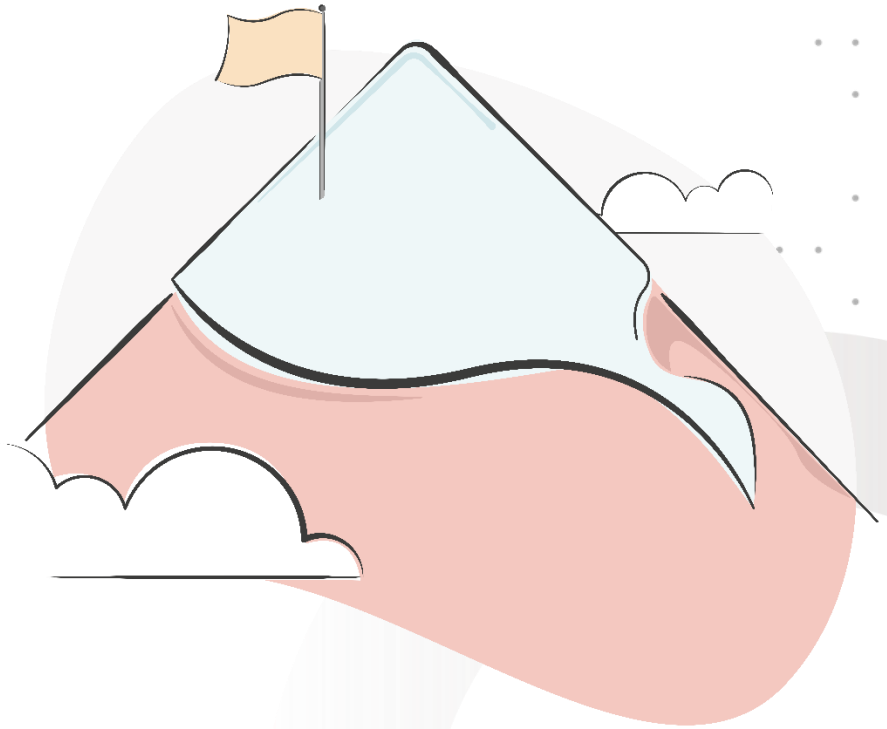
BPP

Session aim and objectives



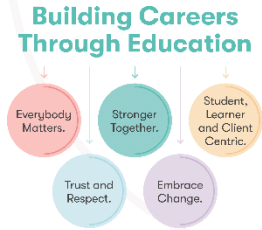
By the end of this session, you should be able to:

1. Explain the fundamentals of event-driven architectures.
2. Understand the pub-sub model, messages, and topics.
3. Explore event streaming with Kafka, including consumer groups, partitions, and clusters.
4. Develop effective collaboration and presentation skills within data engineering projects.



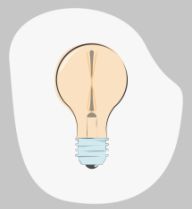
BPP

The evolution of data processing

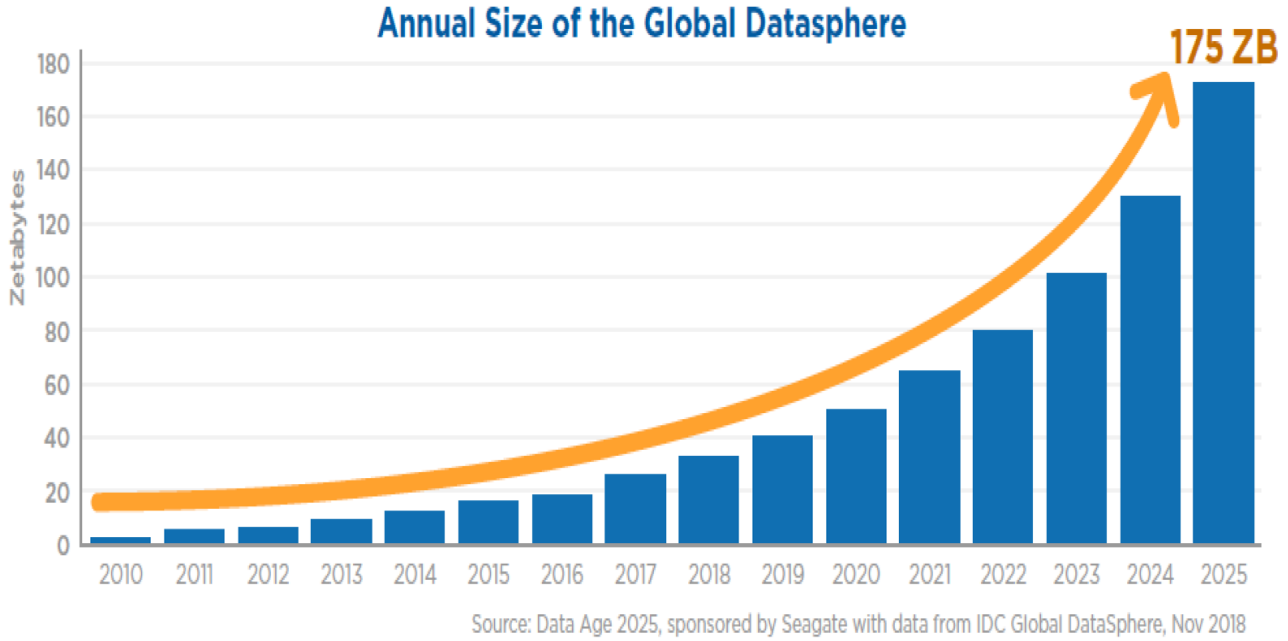


Leading to a more dynamic approach

- Vast data generation every second
- Global data sphere to reach 175 zettabytes by 2025 (IDC)
- Traditional batch processing struggles with real-time insights
- Event-Driven Architectures: dynamic and scalable solution



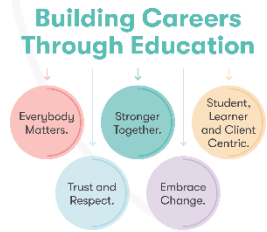
175 zettabytes is equivalent to a stack of DVDs that could reach the moon 23 times or circle the Earth 222 times.



An illustration of data creation growth projections.
Image source: Forbes.com

BPP

Fundamentals of Event-Driven architectures



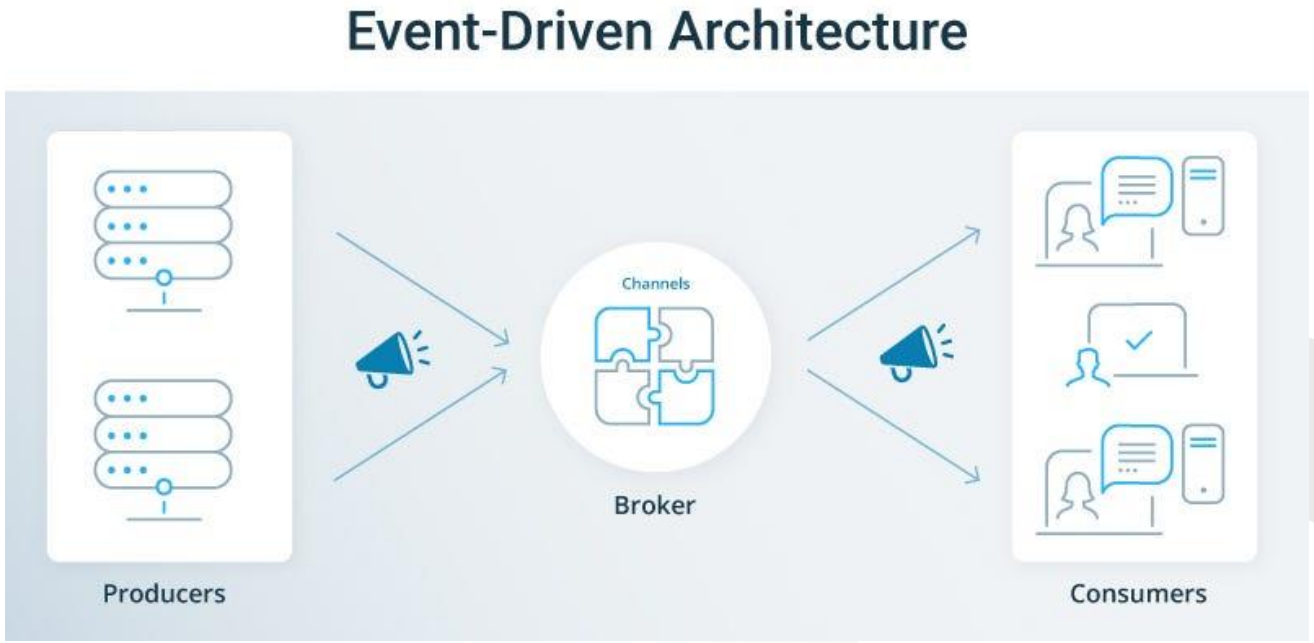
Definition and key components

Design paradigm: Components communicate through events

Asynchronous communication: Enhances responsiveness and scalability

Key Components:

- **Events**: Significant state changes (e.g., purchase, donation, sensor detection)
- **Event producers**: Generate events (e.g., online banking app)
- **Event consumers**: Process events (e.g., fraud detection system)
- **Event brokers**: Route events, ensure scalable and reliable delivery



An illustration of an Event-Driven Architecture
Image source: Medium

Industry relevance of EDA

Building Careers Through Education

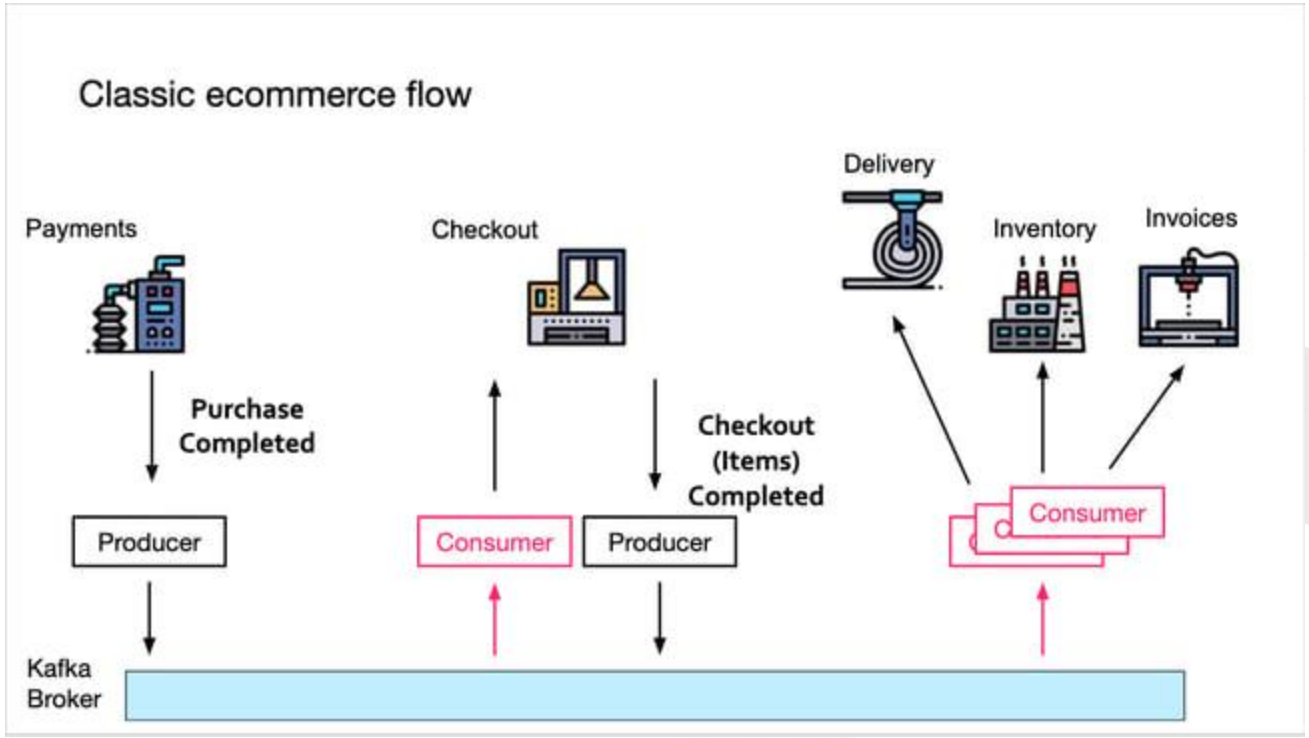
Everybody Matters. Stronger Together. Student, Learner and Client Centric. Trust and Respect. Embrace Change.

Discussion

EDAs are used in a variety of industries to deliver business value. Some examples include:

- Healthcare
- Retail
- Transportation

How would you expect EDAs to be used in these industries?



A classic EDA from ecommerce
Image source: Hubspot

BPP

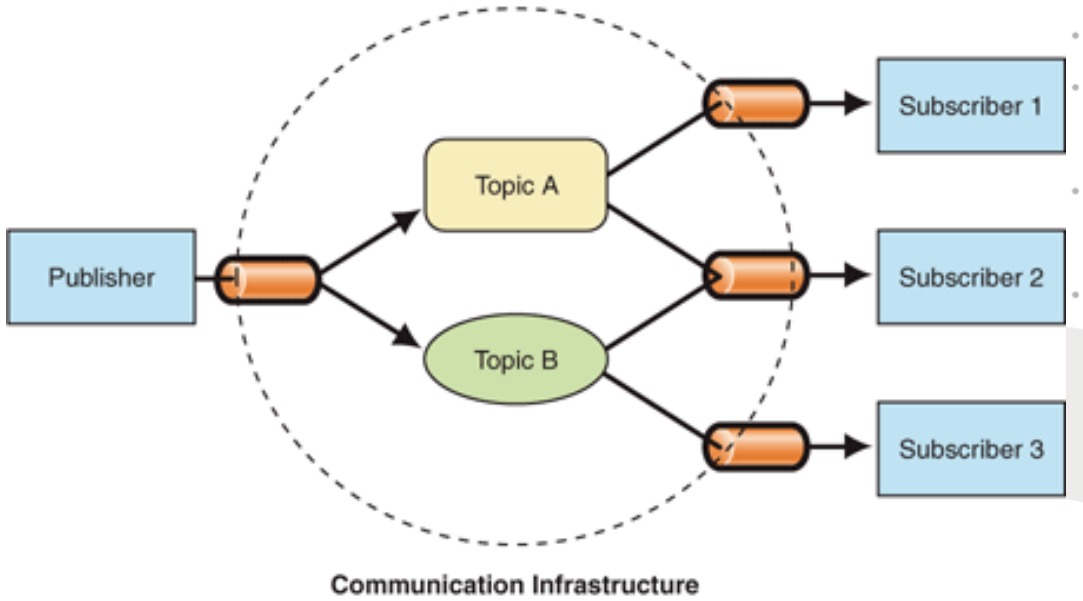
The Publish-Subscribe (Pub-Sub) model

Explanation and advantages…

- **Messaging Pattern**: Central to Event-Driven Architecture (EDA)
- **Publishers**: Emit messages to specific topics
- **Subscribers**: Receive messages based on their interest in topics
- **Decoupling**: Publishers and subscribers operate independently

An analogy - Stock market data feeds:

Financial news services (publishers) provide real-time stock prices. Investors and trading systems (subscribers) receive this data to make informed decisions, without direct interaction.

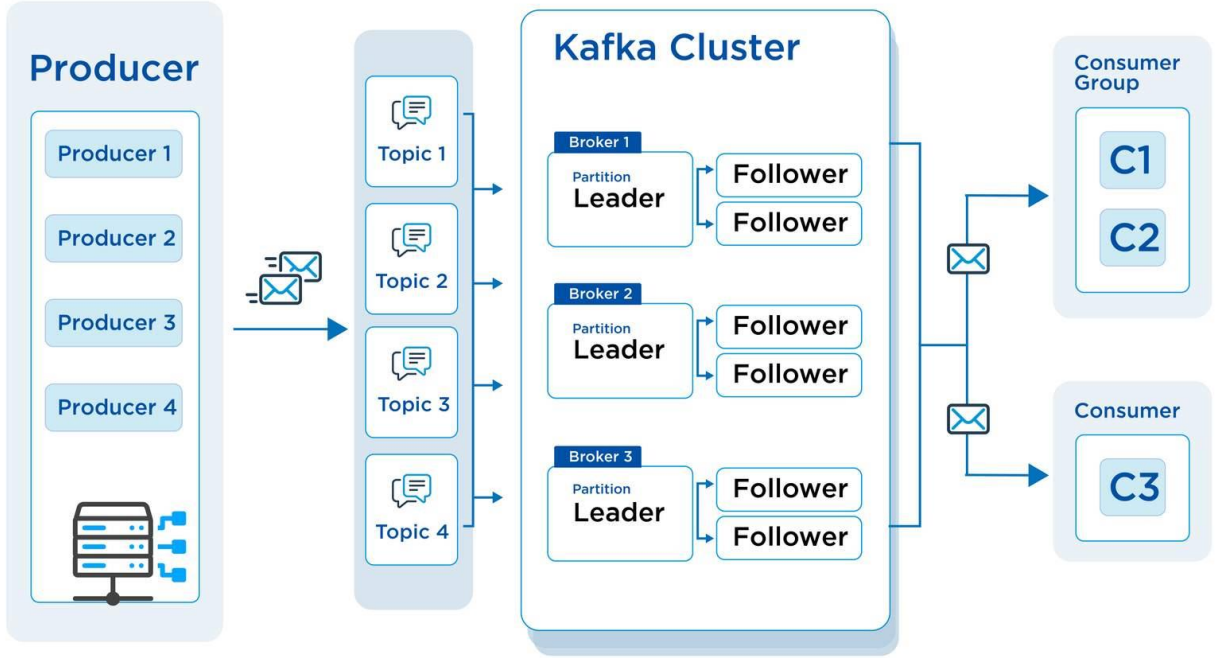


An example of a Publisher-Subscriber pattern
Image source: liliendahl

Introduction to Kafka

Kafka and its significance in EDA

- **Topics**: Categories for messages (e.g., "transaction_events", "user_activity")
- **Partitions**: Subdivisions of topics for parallel processing
- **Producers**: Applications writing data to topics
- **Consumers**: Applications reading data from topics
- **Consumer Groups**: Collections of consumers ensuring load balancing and fault tolerance



An illustration of the Kafka ecosystem
Image source: LinkedIn

Kafka Clusters and consistency

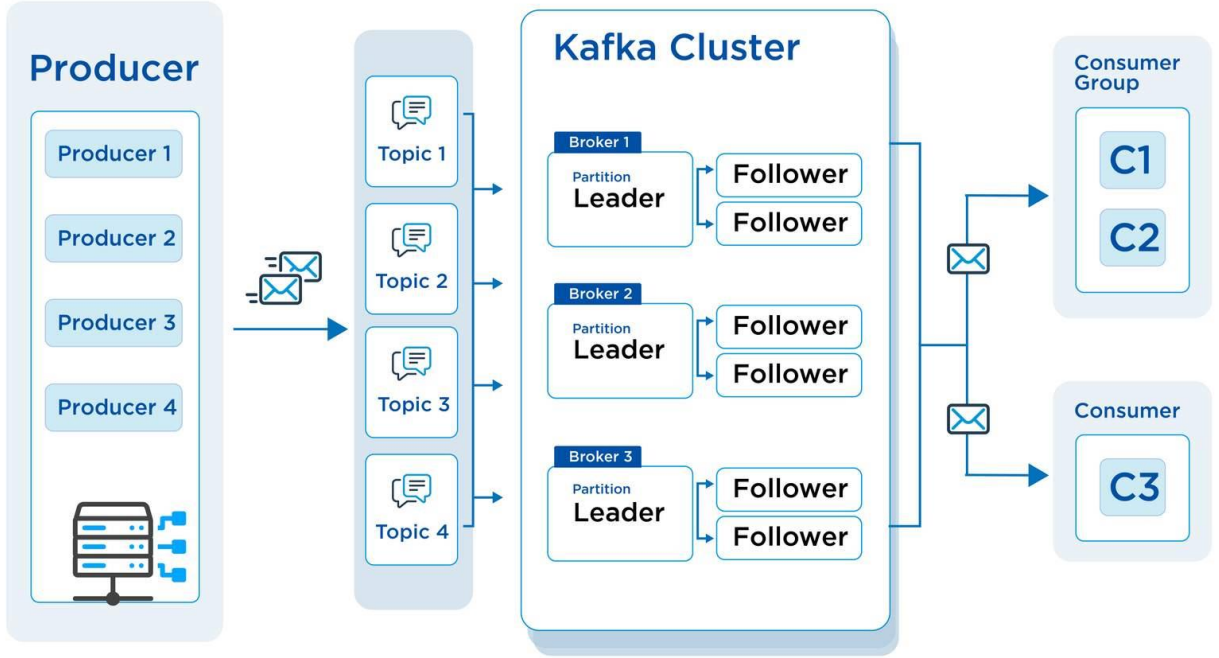
Kafka clusters, brokers, and data consistency

Kafka Cluster:

- Multiple brokers manage storage and retrieval
- Zookeeper ensures consistency and reliability
- Scales horizontally by adding brokers

Data Consistency:

- Replication: multiple replicas across brokers
- Leader and follower brokers for each partition
- Strong consistency: all replicas acknowledge messages
- Ensures availability and consistency even if a broker fails

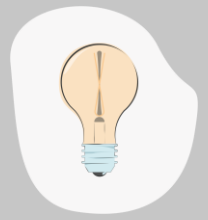


An illustration of the Kafka ecosystem
Image source, LinkedIn: Link

Kafka as a Log

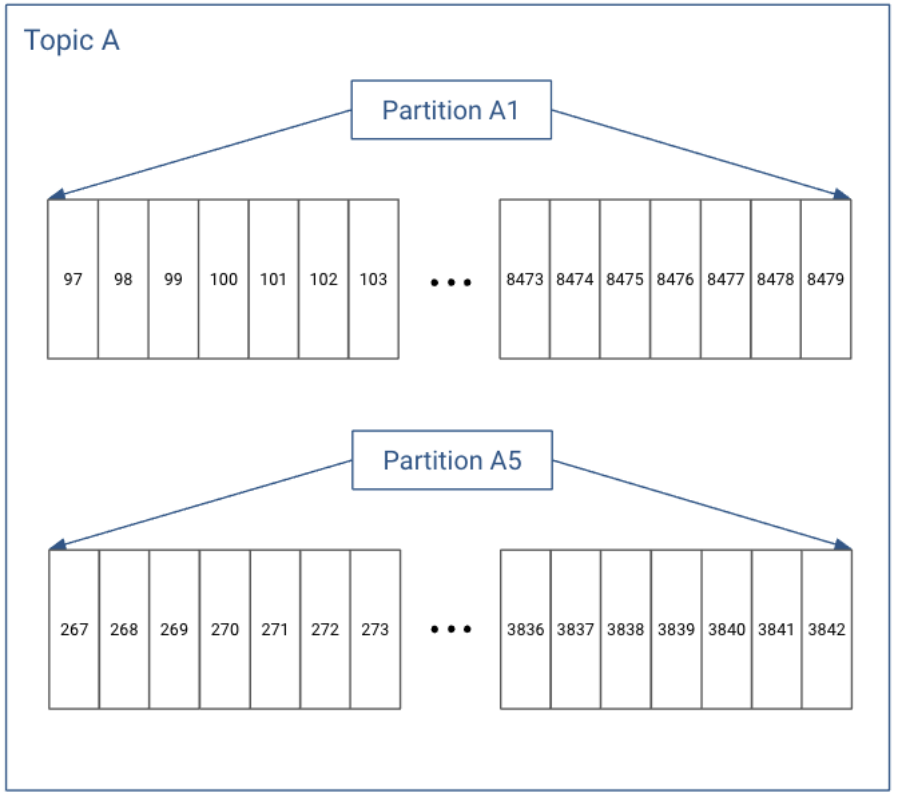
Kafka's log-based architecture and its benefits

- Partitions as immutable logs
- Messages appended sequentially with offsets
- Consumers track positions via offsets
- Efficient handling of large data volumes
- Durability through configurable retention periods



Configuration considerations:

- Retention policies
- Replication factor
- Producer and consumer settings



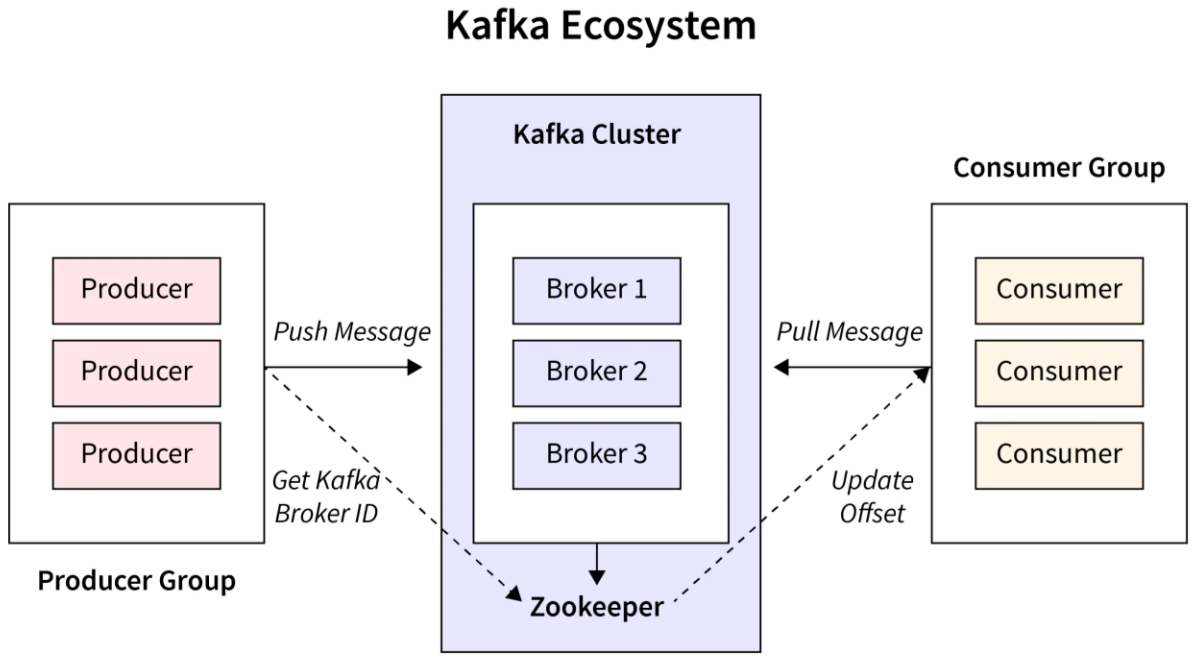
A diagram illustrating the principles of effective solution design
Image source: Cloudera

Kafka setup issues

You may encounter some challenges:

- Zookeeper coordination
- Network configuration
- Resource allocation
- Security configuration

These issues will be explored during the practical lab for this webinar.



An illustration of the Kafka ecosystem including Zookeeper
Image Source Scaler Topics

Effective collaboration in data engineering

Key techniques

Clear Communication:

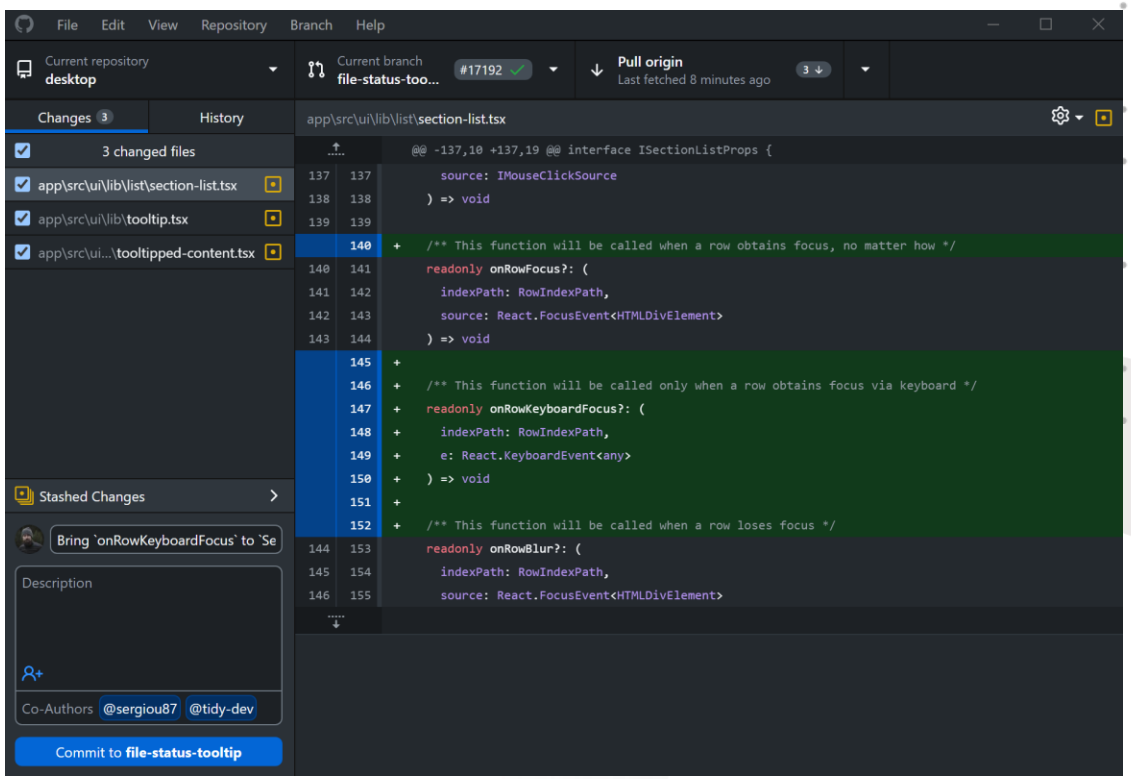
- Use tools like Slack or Microsoft Teams
- Maintain documentation in shared repositories (GitHub, Confluence)

Defined Roles and Responsibilities:

- Assign specific tasks (data modeling, pipeline development, testing)
- Rotate roles to develop diverse skills

Regular Stand-Ups and Meetings:

- Daily or weekly progress updates
- Address blockers
- Use Agile methodologies



An example of a GitHub repository
Image source: [GitHub]

BPP

Presenting with Kafka

Effective techniques

Effective techniques for presentation and storytelling

Time for the practical lab!

Your tutor will provide guidance as required…

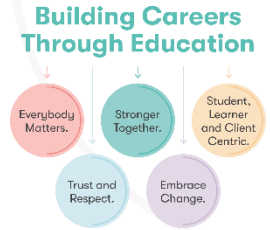
Practical lab activities detailed in this document: Lab activities

Key Learning Summary

- **EDA**: Event-Driven Architectures enable asynchronous communication, enhancing system responsiveness and scalability.
- **Pub-Sub Model**: Publishers send messages to topics; subscribers receive messages based on interest, allowing decoupled communication.
- **Apache Kafka**: An event streaming platform used by over 80% of Fortune 100 companies for high-throughput, fault-tolerant data streams.
- **Kafka Core Concepts**: Includes topics, partitions, producers, consumers, and consumer groups for efficient data processing.
- **Kafka Cluster:** Multiple brokers managed by Zookeeper ensure data consistency and scalability.
- **Data Consistency**: Kafka uses replication with leader and follower brokers to ensure strong consistency and availability.
- **Log-Based Architecture**: Kafka treats partitions as immutable logs, efficiently handling large data volumes with configurable retention.

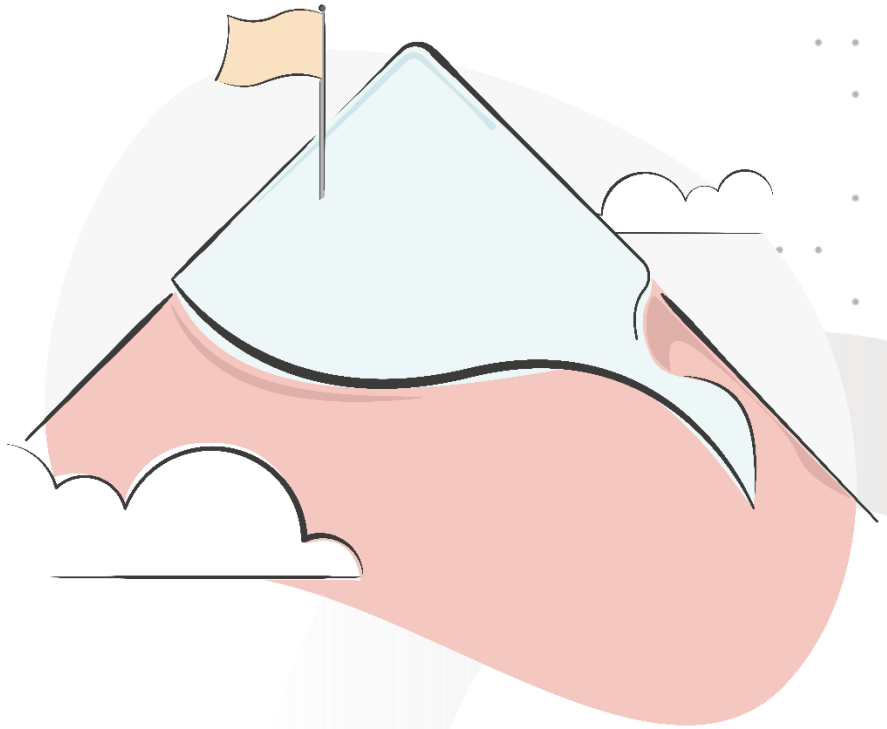
BPP

Session aim and objectives



You should now be able to:

1. Explain the fundamentals of event-driven architectures.
2. Understand the pub-sub model, messages, and topics.
3. Explore event streaming with Kafka, including consumer groups, partitions, and clusters.
4. Develop effective collaboration and presentation skills within data engineering projects.



BPP

Thank you

Do you have any questions, comments, or feedback?