

# Big Data Technologies and Architectures

@Tekna --- May 25-26

Marc Gallofré Ocaña

marcgallofre@gmail.com



/marcgallofre

Ga11u



# Who am I?



**Name:** Marc Gallofré Ocaña



**Origin:** Barcelona (but currently living in Bergen)



**Position:** Researcher



**Sector:** AI systems in newsrooms

# More about me

- ▶ **PhD. in AI systems and software architectures** – University of Bergen
- ▶ **MSc. in Innovation and Research in Informatics (Big Data)** – Polytechnic University of Catalonia-BarcelonaTech
- ▶ **BSc. in Informatics Engineering (Information Systems)** – Polytechnic University of Catalonia-BarcelonaTech
- ▶ **Worked as Business Intelligence consultant in Barcelona for different sectors (sports, real state, emergencies).**

# Who are you?



**Name:**



**Origin:**



**Position:**



**Sector:**



**Why did I join the workshop?**



**Previous knowledge on the topic**

# Agenda

## ▶ Thursday

- ▶ Big data and industry
- ▶ Architectures I
- ▶ Brainstorming and discussion
- ▶ Handling Data
- ▶ Streaming with Kafka
- ▶ Hands-on Kafka

## ▶ Friday

- ▶ NoSQL DBs
- ▶ Hands-on MongoDB
- ▶ Hands-on Cassandra
- ▶ Architectures II
- ▶ Brainstorming and discussion
- ▶ Wrap-up

# Time to activate our brains



# Agenda

- ▶ What is Big Data?
- ▶ What implications can Big Data have?
- ▶ How does Big Data relate to AI and Knowledge Graphs?
- ▶ Examples of Big Data across industries
- ▶ Architecture patterns I

# Task: True or False

- ▶ Build groups.
- ▶ Decide who is the facilitator and spokesperson.
- ▶ Decide whether each statement is true or false and justify the decision with facts, examples or experiences (15')
- ▶ Share and discuss.



# Statements

- ▶ Big data can only be collected from online sources such as websites and social media.
- ▶ Machine learning algorithms are always superior to traditional statistical methods in big data analysis.
- ▶ Data privacy concerns do not apply to anonymized big data.
- ▶ Implementing big data technologies guarantees immediate and substantial ROI for all businesses.
- ▶ In 2022, the total amount of data created worldwide was estimated to be 97 zettabytes.
- ▶ Big data analysis eliminates the need for human involvement in decision-making processes.

# Statements

- ✗ Big data can only be collected from online sources such as websites and social media.
- ✗ Machine learning algorithms are always superior to traditional statistical methods in big data analysis.
- ✗ Data privacy concerns do not apply to anonymized big data.
- ✗ Implementing big data technologies guarantees immediate and substantial ROI for all businesses.
- ✓ In 2022, the total amount of data created worldwide was estimated to be 97 zettabytes (97.000.000.000.000 GB)
- ✗ Big data analysis eliminates the need for human involvement in decision-making processes.

# Statements

**✗ Big data can only be collected from online sources such as websites and social media.**

# Statements

**✗ Big data can only be collected from online sources such as websites and social media.**

- Explanation: Big data can be collected from various sources, including online and offline channels. Offline sources can include sensor data, IoT devices, customer interactions in physical stores, and more.

# Statements

**✗ Machine learning algorithms are always superior to traditional statistical methods in big data analysis.**

# Statements

- ✗ **Machine learning algorithms are always superior to traditional statistical methods in big data analysis.**
  - Explanation: While machine learning algorithms are powerful for big data analysis, traditional statistical methods can still be effective and have their own advantages depending on the specific context and requirements.

# Statements

**✗ Data privacy concerns do not apply to anonymized big data.**

# Statements

**✗ Data privacy concerns do not apply to anonymized big data.**

- ▶ Explanation: Even with anonymized data, there can still be privacy concerns, as certain pieces of information can be combined or inferred to re-identify individuals. Robust privacy measures need to be implemented to protect individuals' privacy rights.



# Statements

**✗ Implementing big data technologies guarantees immediate and substantial ROI for all businesses.**

# Statements

**✗ Implementing big data technologies guarantees immediate and substantial ROI for all businesses.**

- Explanation: Implementing big data technologies requires careful planning, proper infrastructure, and skilled resources. The return on investment (ROI) may vary depending on various factors, including the organization's goals, use case, and implementation strategy.

# Statements

✓ In 2022, the total amount of data created worldwide was estimated to be 97 zettabytes (97.000.000.000.000 GB)

# Statements

✓ In 2022, the total amount of data created worldwide was estimated to be 97 zettabytes (97.000.000.000.000 GB)

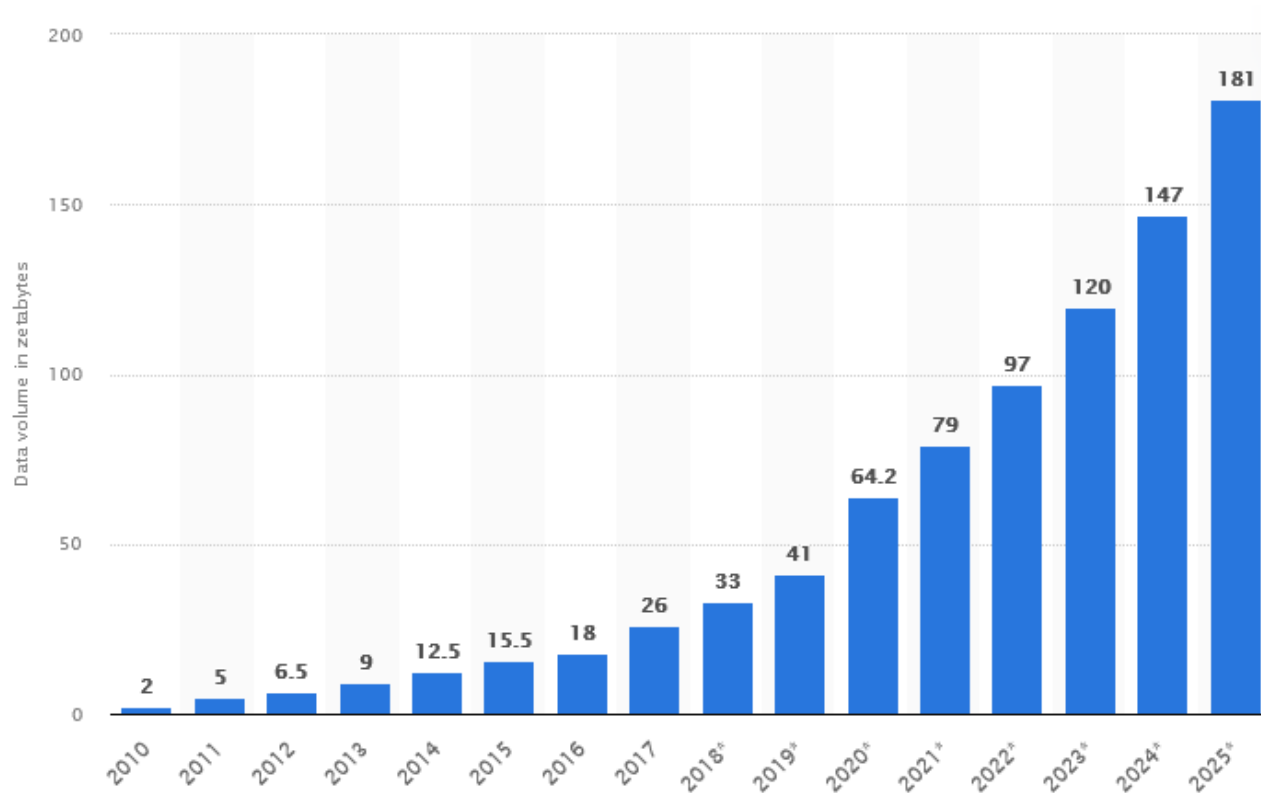


Figure from: <https://www.statista.com/statistics/871513/worldwide-data-created/>

# Statements

- ✗ Big data analysis eliminates the need for human involvement in decision-making processes.**

# Statements

**✗ Big data analysis eliminates the need for human involvement in decision-making processes.**

- Explanation: Big data analysis enhances decision-making processes by providing data-driven insights. However, human involvement is still crucial in interpreting the results, considering ethical considerations, and applying domain knowledge to make informed decisions. Big data is a tool that complements human decision-making, rather than replacing it entirely.

# What is Big Data?



# What is Big Data?

Big Data refers to the vast and complex volumes of data generated at high speeds from diverse sources.



# Big Data Dimensions

1. Volume
2. Velocity
3. Variety
4. Variability
5. Validity
6. Veracity
7. Value

# Big Data Dimensions

1. **Volume:** large volumes of data that exceed the capacity of traditional data processing systems.
2. **Velocity:** it represents the speed at which data is generated and needs to be processed in real time or near real time.
3. **Variety:** it refers to the diverse types and formats of data.

## Other dimensions

4. **Variability**: data can vary in terms of structure, format, and quality over time.
5. **Validity**: data must be accurate, correct and relevant.
6. **Veracity**: the trustworthiness of the data.
7. **Value**: how good is the data for the objectives or goals.

# Big Data Implications

We know what Big Data is, but what about the risks?

# The TARGET case (TARGET is USA retail store)

“My daughter got this in the mail!” he said. “She’s still in high school, and you’re sending her coupons for baby clothes and cribs? Are you trying to encourage her to get pregnant?”

Context: An email was addressed to the man’s daughter and contained advertisements for maternity clothing, nursery furniture and pictures of smiling infants.

<https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/>

# The TARGET case

- ▶ TARGET identified the man's daughter as a pregnant woman based on her recent shopping and search history.
- ▶ Indeed, the daughter was pregnant but neither the father nor the daughter knew it.
- ▶ TARGET discovered private information from the daughter by analyzing data and collecting patterns from a large number of clients.

# Big Data privacy

- ▶ **Data privacy:** concerns arise when personal and sensitive information is collected and processed without proper consent, inadequate anonymization or in violation of privacy regulations from various sources, such as social media, bank transactions and internet browsing.
- ▶ **“Unintentional” Identification:** During the analysis process, patterns or correlations may emerge that unintentionally reveal personal information or allow individuals to be identified. This could happen through data linkage, cross-referencing with external sources, or sophisticated data mining techniques.

# Big Data privacy

- ▶ **Anonymization and De-identification:** these techniques can remove or mask personal identifiers from the dataset. This helps minimize the risk of re-identification and unauthorized access to individuals' personal information.
- ▶ **Supervised decision-making:** including the human decision as part of the process and outcomes of big data processing.



# Big Data and Society

The use of Big Data analytics raises ethical questions regarding the appropriate use and handling of data. This includes concerns about data ownership, data sovereignty, algorithmic bias, discrimination, and the potential misuse of insights derived from Big Data.

# Big Data, AI and Knowledge Graphs

The background of the slide features an abstract geometric design. On the right side, there are several overlapping triangles in various shades of teal and grey, creating a dynamic, layered effect. The rest of the slide has a plain, light grey background.

# Big Data and AI

Training AI models require large amounts of data that need to be collected, prepared and processed.

- ▶ GPT-3 -> 54TB of text data
- ▶ LLaMA -> 4.749TB of text data
- ▶ Stable Diffusion -> 5.85 billion pairs of high-quality images and text (>600TB)

GPT: <https://openai.com/product/gpt-4>

LLaMA: <https://ai.facebook.com/blog/large-language-model-llama-meta-ai/>

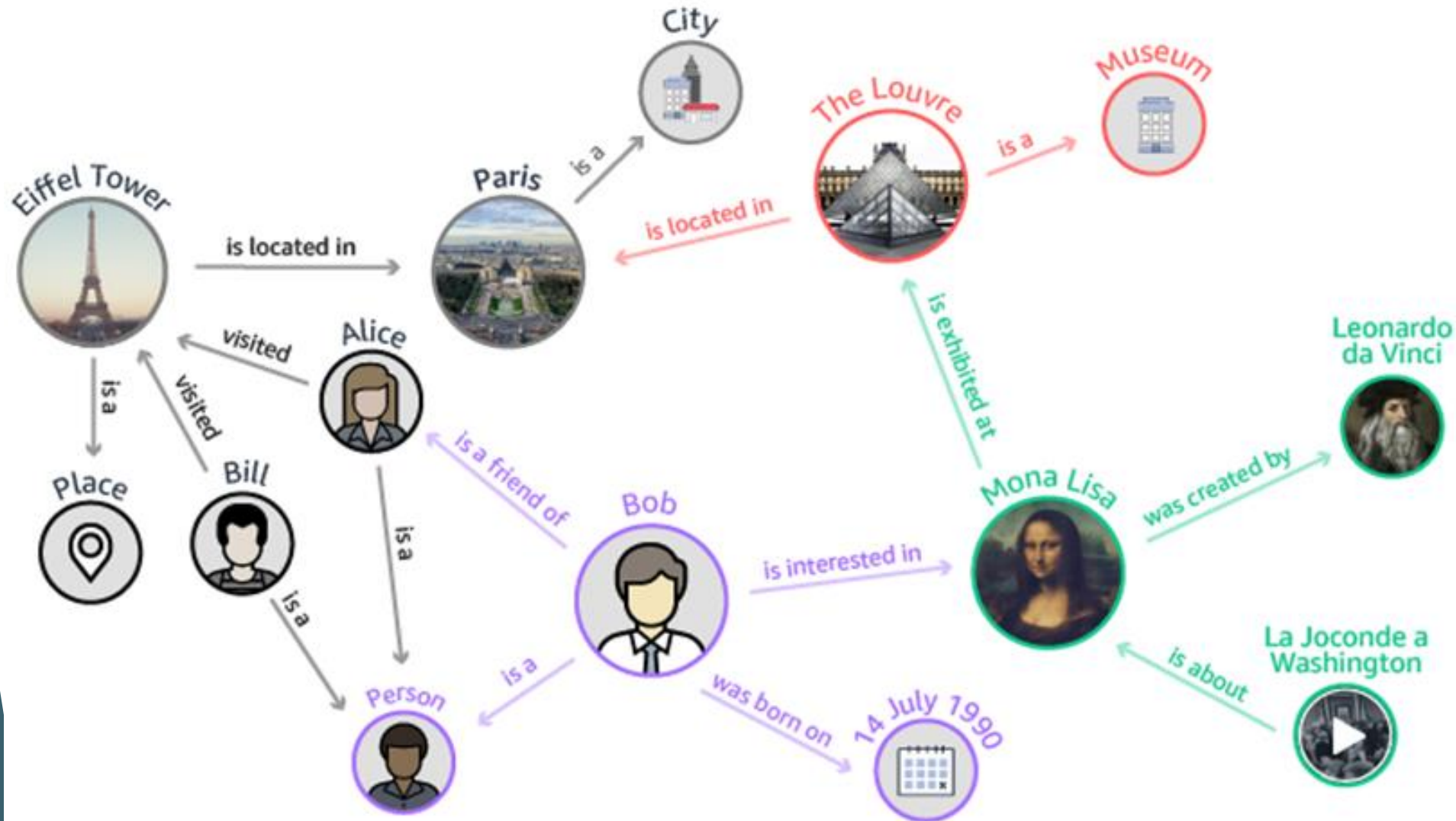
Stable Diffusion: <https://stability.ai/stable-diffusion>

# Big Data and AI

But also keeping up-to-date the models:

- ▶ Intelligent cars are constantly producing data to improve the models.
- ▶ Recommendation systems in retail and online shops.
- ▶ Stock market prediction models

# Big Data and Knowledge Graphs



# Big Data and Knowledge Graphs

Some examples of Knowledge Graphs:

- ▶ UK legislation ([data.gov.uk](http://data.gov.uk))
- ▶ Felles datakatalog ([data.norge.no](http://data.norge.no))
- ▶ Google Knowledge Graph (8 billion entities)
- ▶ Dbpedia: 7.14 Billion triples (“rows” of a database)
- ▶ Wikidata: 14.7 Billion triples

# Big Data in Industry

The background features abstract geometric shapes in various shades of teal, blue, and grey, creating a modern, data-oriented aesthetic. The shapes are layered and angular, with some appearing as large triangles and others as smaller, overlapping polygons. The overall composition is clean and professional, suitable for a corporate or technical presentation.

# Examples of Big Data across industries

**Healthcare providers** use Big Data to analyze large volumes of patient data, including electronic health records, medical images, and genetic information, to identify patterns and correlations. This helps in early disease detection, personalized medicine, and treatment optimization. Big Data also supports population health management by analyzing health trends and predicting disease outbreaks.



# Examples of Big Data across industries

**Intelligent Traffic Management:** Big Data analytics is used to manage and optimize traffic flow. By collecting and analyzing data from various sources such as traffic cameras, sensors and GPS devices, transportation authorities can gain real-time insights into traffic patterns, congestion hotspots, and road conditions. This data helps in implementing intelligent traffic management systems, optimizing signal timings, and improving overall transportation efficiency.

# Examples of Big Data across industries

The **oil industry** uses Big Data analytics to implement predictive maintenance strategies in oil refineries. By leveraging sensor data, historical maintenance records, and real-time monitoring systems, companies can analyze patterns and identify potential equipment failures or maintenance needs in advance. This allows them to schedule maintenance activities proactively, reduce downtime, optimize operational efficiency, and minimize costly unplanned shutdowns.

Any example in your industry?



# Architectures I

# What do they have in common with Twitter?



August 3, 2013

# What do they have in common with Twitter?



August 3, 2013

How many tweets are sent every day?

# How many tweets are sent every day in avg?

- ▶ A) less than 1 million
- ▶ B) more than 500 million
- ▶ C) between 10 and 50 million
- ▶ D) around 100 million



“New Tweets per second (TPS) record: 143,199 TPS.  
Typical day: more than 500 million Tweets sent;  
average 5,700 TPS.”

- Raffi Krikorian, VPE Twitter, 2013

**Recommended lecture:** [https://blog.twitter.com/engineering/en\\_us/a/2013/new-tweets-per-second-record-and-how](https://blog.twitter.com/engineering/en_us/a/2013/new-tweets-per-second-record-and-how)

What may happen to a non-big data system when it reaches these volumes?

# What may happen to a **non**-big data system when it reaches these volumes?

- ▶ A) Nothing, systems are made to handle large amounts of data.
- ▶ B) Systems may crash, become unavailable and not respond.
- ▶ C) The engineering team has to work overnight to fix the problems.
- ▶ D) The system needs to be redesigned with new architecture and technology.

# What happens when data goes big?

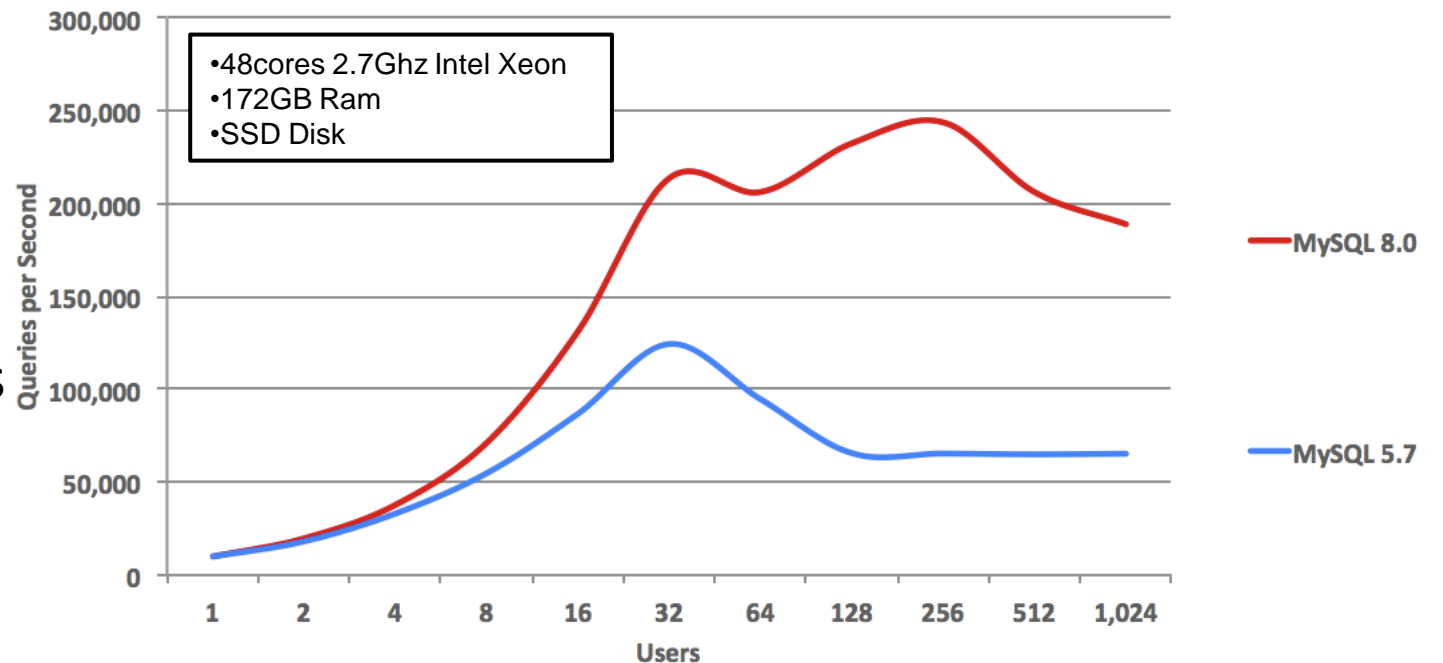
## MySQL limits:

- Tablespace: 256TB
- Row limit: 65536 bytes
- Max of ca. 3-4 Billion ( $10^9$ ) records

## Twitter numbers:

- 500 M tweets x 7 days = 3.5 B
- It crashes in a week

Benchmark of RW queries on a 10M table



<https://dev.mysql.com/doc/refman/8.0/en/innodb-limits.html>

[http://dimitrik.free.fr/blog/posts/mysql-performance-80-and-sysbench-oltp\\_rw-updatenokey.html](http://dimitrik.free.fr/blog/posts/mysql-performance-80-and-sysbench-oltp_rw-updatenokey.html)

# What happens when data goes big?

Benchmark of RW queries on a 10M table

MySQL limits

- Tablespace
- Row limit
- Max of ca

**Limits exist beyond databases.**

**Computational resources like  
Memory, CPU, GPU have limits too!**

— MySQL 8.0

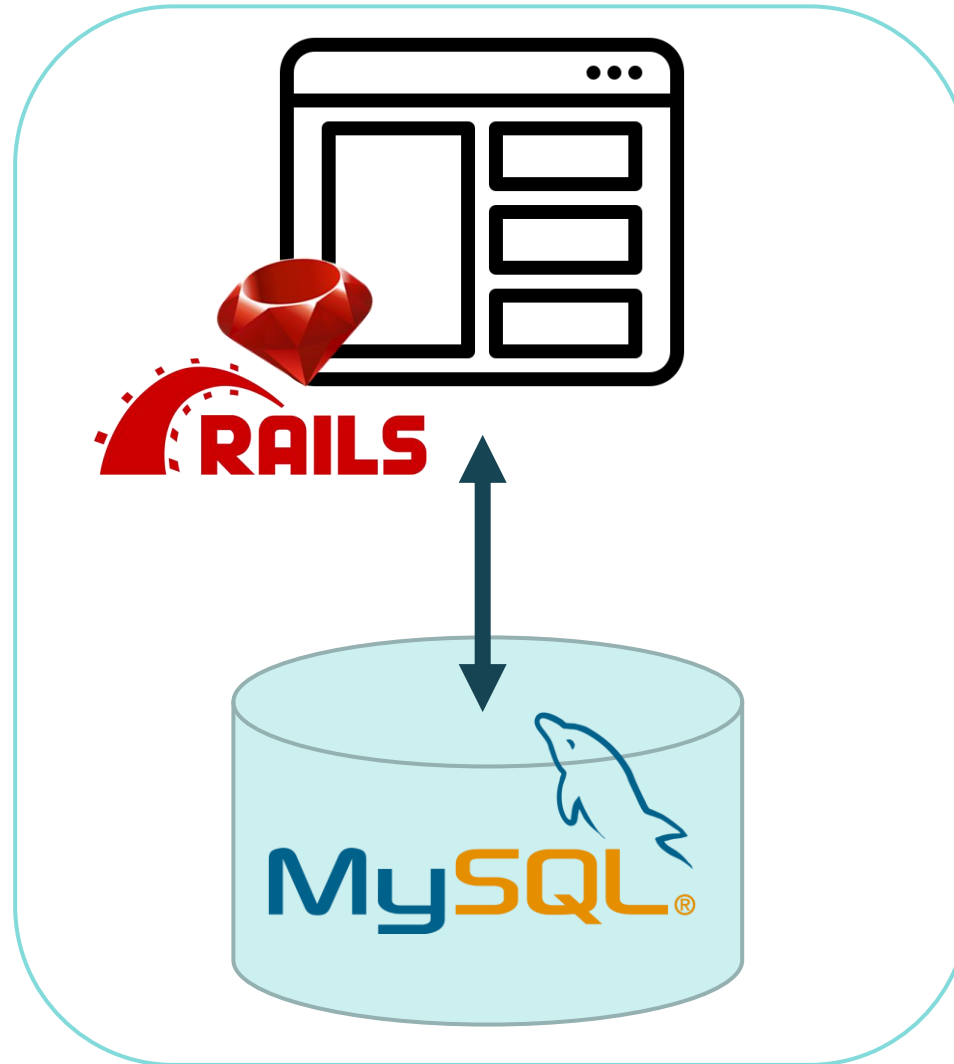
— MySQL 5.7

Twitter

- 500
- It crashes in less than week

<https://dev.mysql.com/doc/relnotes/8.0/en/mysql-8.0-new-features.html>  
[http://dimitrik.free.fr/blog/posts/mysql-performance-80-and-sysbench-oltp\\_rw-updatenokey.html](http://dimitrik.free.fr/blog/posts/mysql-performance-80-and-sysbench-oltp_rw-updatenokey.html)

# How did Twitter look like on that time?



# Monolithic architecture

- ▶ The whole system is written in a single codebase.
- ▶ Difficult to horizontally scale.
- ▶ Easy to deploy and vertical scale.

What can you do you improve it?

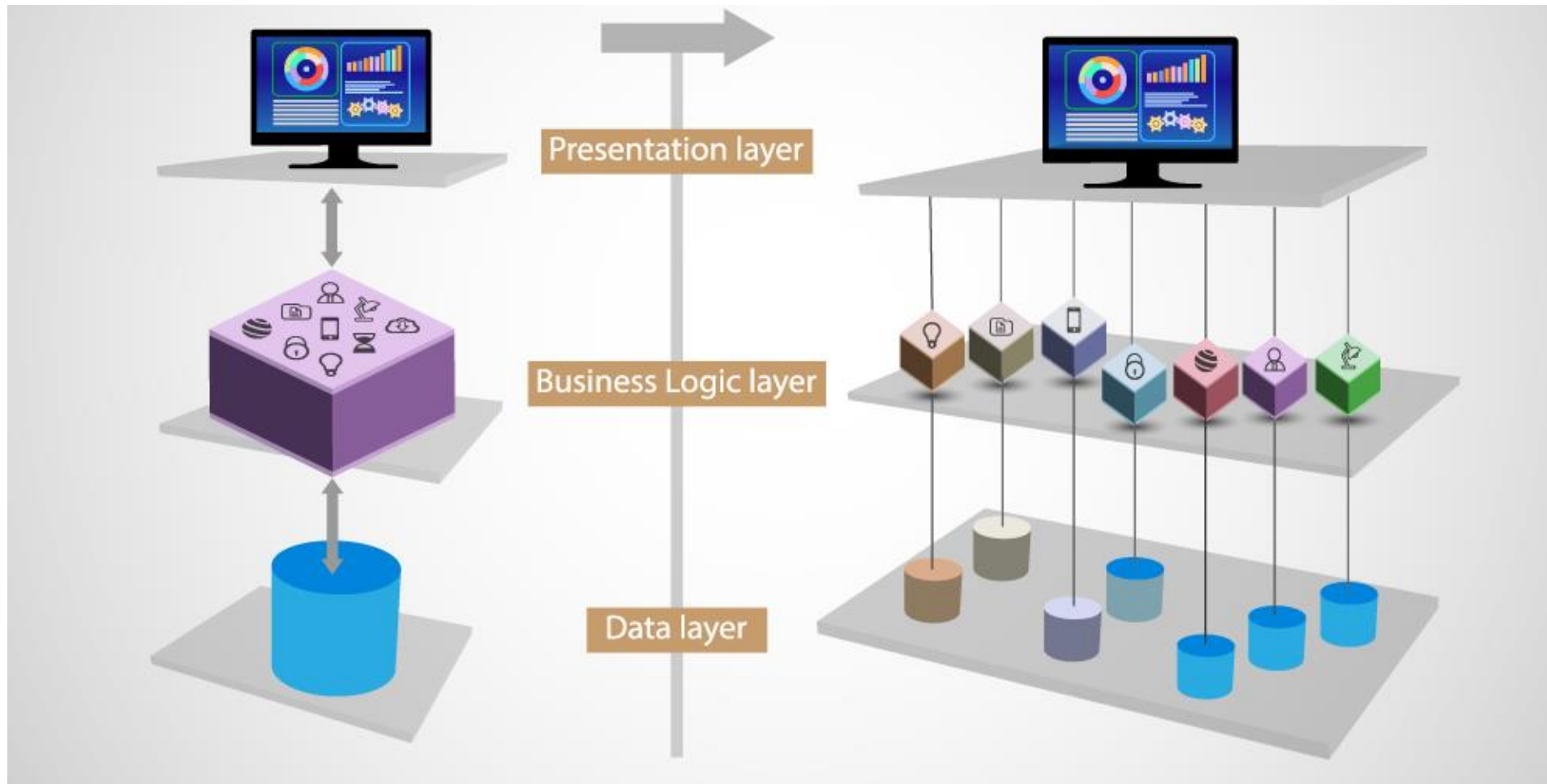




**Divide  
and  
Conquer!**



# Microservice



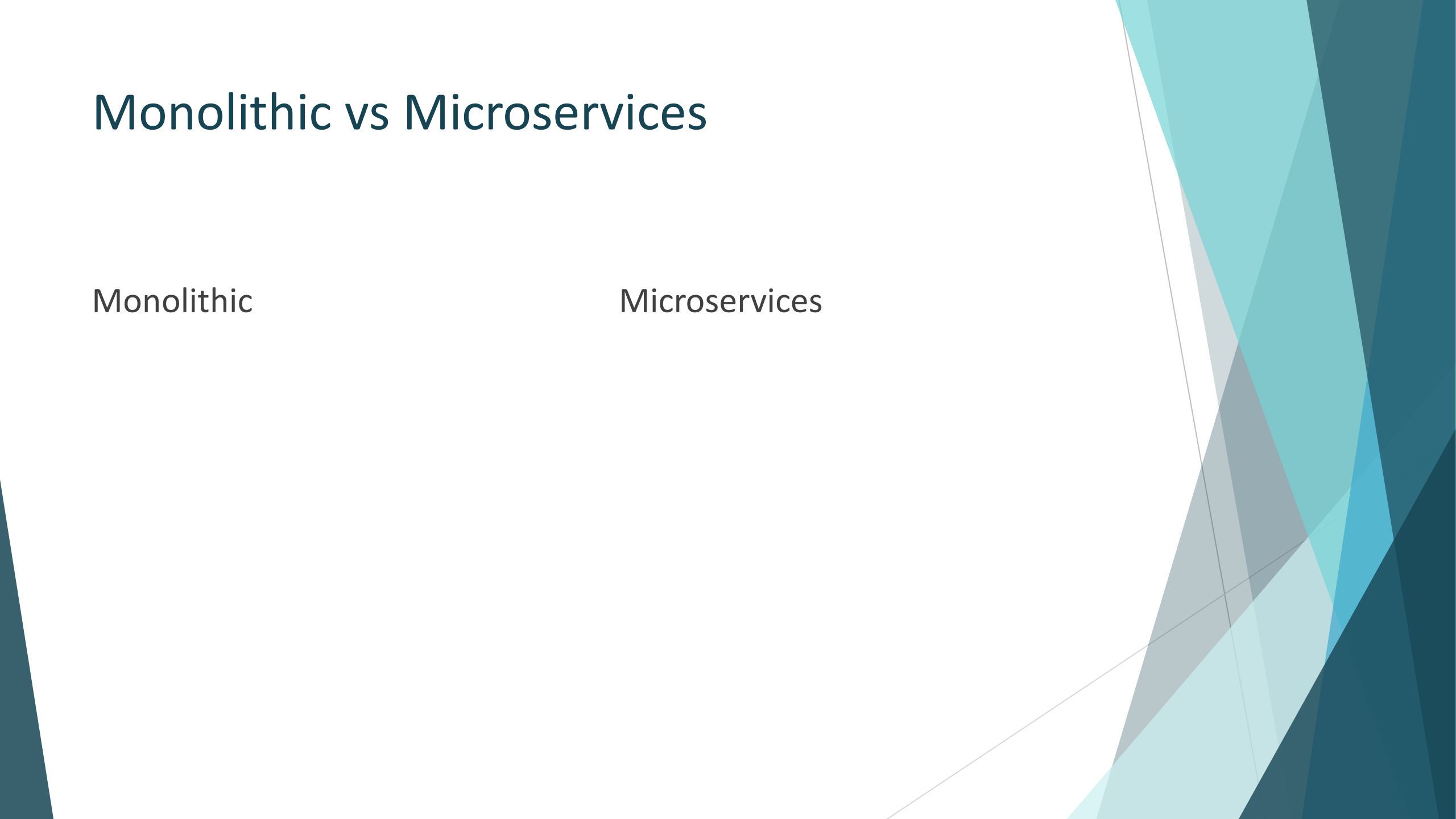
# Microservice architecture

- ▶ The system is composed of loosely coupled services.
- ▶ Services are modular and easy to replicate.
- ▶ Services have clear boundaries and APIs.
- ▶ Services are **small** and typically implement a function of the system

# Monolithic vs Microservices

Monolithic

Microservices



## Monolithic

- ▶ Easier and faster to deploy, all code in the same codebase.
- ▶ Difficult to scale and distribute.
- ▶ Difficult to maintain.
- ▶ Tightly coupled processes.
- ▶ May be good for small applications.
- ▶ Use a single technology stack which can difficult the integration.
- ▶ Easier to secure.
- ▶ An error effect the whole system.

## Microservices

- ▶ More effort to develop.
- ▶ Easier to distribute and scale.
- ▶ Easier to maintain.
- ▶ Lousy coupled components.
- ▶ Can easily combine and integrate different technologies.
- ▶ Many independent components and APIs that need to be secured.
- ▶ An error effect one service.

# Service-oriented architecture (SOA)

- ▶ Similar to microservices, but the services are larger and implement business functionalities.
- ▶ One service can implement and group multiple functions together.

# Serverless architecture

- ▶ The system is conceived as functions rather than services.
- ▶ These functions are deployed as small units of code
- ▶ The functions typically run on an event-based infrastructure from a cloud provider that handles the resource provisioning.



Payment Service

SOA

Order Management,  
Refund Management

Microservices

Sent Email, Accept payment, Verify  
payment, refund payment

Serverless



# Brainstorming on Big Data

### Part 1:

- ▶ Each member of the team writes down a potential or current application of Big Data (it can be in your industry, field or something else) (3')
- ▶ Pass the idea to the member of your team on your left side.
- ▶ This person has 2 minutes to complement it.

### Part 2:

- ▶ You got an idea from someone else.
- ▶ Write one vulnerability, obstacle, risk or problem (2').

### Part 3:

- ▶ Your group gathers the ideas, pick one idea and discuss their potential and how the risk can be addressed. (5')
- ▶ Present and share the conclusions and debate.

# Handling Big Data

# Data Snapshot

1. You have 10 minutes to create a visual representation of a snapshot of big data in your industry. This can include data sources, data collection methods, data analysis techniques, or any relevant aspects associated with big data.
2. You have 2-3 minutes to explain and share your visual representation and share your thoughts on the significance of big data in your context.

# Big Data Challenges

Which problems can the traditional systems or analysis approaches have with Big Data?

# Big Data Challenges

- ▶ Difficulties in storing, managing, and processing big data
- ▶ Accommodating diverse data types and structures
- ▶ Storage limitations and costs
- ▶ Ensuring availability, reliability, and consistency
- ▶ Traditional data processing struggles with the scale of big data
- ▶ Efficient processing for real-time data
- ▶ Performance bottlenecks and latency
- ▶ Scalability for growing data volume and velocity

# Big Data technologies

Big data analysis  
frameworks

Data streaming

Real-time  
analysis

Big Data storing

# Big data analysis frameworks

There are different frameworks for processing and analysing large-scale data with the goal of facilitated distributed computing and parallel processing.

► Examples: MapReduce, Hadoop, Spark, Storm

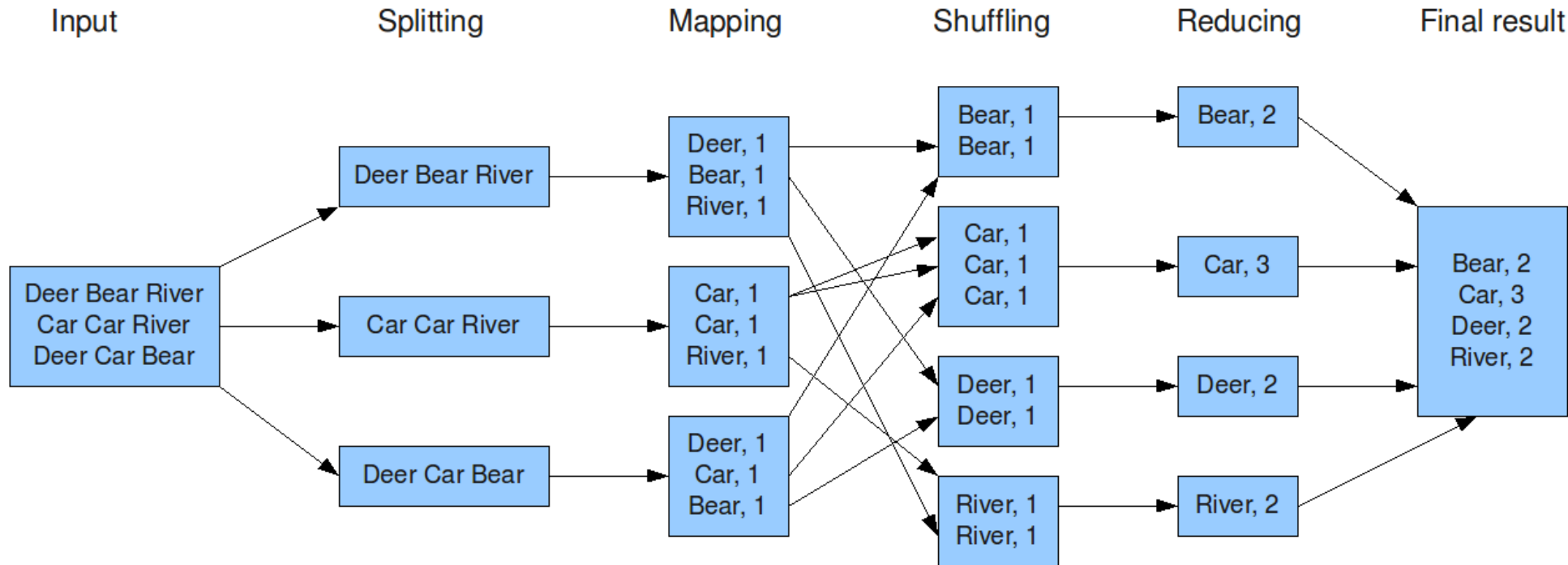


# MapReduce

MapReduce is a programming model and processing framework designed for processing and analyzing large-scale data sets in a distributed computing environment. It consists of two main operations: "Map" and "Reduce."

# MapReduce

The overall MapReduce word count process



# Hadoop

Hadoop is an open-source software framework that implements the MapReduce model and provides a distributed file system called Hadoop Distributed File System (HDFS) and a resource management framework (YARN) to manage resources (CPU, memory, etc.).

Hadoop computations are performed on disk.



# Apache Spark

Apache Spark uses a generalization of MapReduce and executes the computations in memory. The spark framework facilitates APIs for machine learning and graph processing over distributed and parallel resources.



# Data Streaming

Data streaming involves the ingestion and handling of continuous data with the goal of handling data in real time.

► Examples: Kafka, RabbitMQ

# Apache Kafka

Apache Kafka is a distributed event streaming platform that follows a publish-subscribe messaging model for handling high-volume data streams. It is highly scalable, fault-tolerant architecture, provides low-latency message delivery, and also distributed storage.



# RabbitMQ

RabbitMQ is a distributed message broker where the messages are pushed to the users and removed once they are consumed. It distributes messages between the users, rather than keeping a log.



# Real-time analysis

Real-time analysis involves analysing data as it is generated in order to gain immediate insights and enable timely actions.

► **Examples** Spark Streaming and Apache Storm.



# Apache Storm

Apache Storm is a distributed real-time computation system. Storm is a real-time approach, whereas Hadoop is for batch processing.



# Apache Spark Stream

Spark Streaming is part of the Apache Spark framework that creates small batches of streaming data (DStream) to be processed by Apache Spark.



# Big Data storing

Big data storing refers to the use of databases specifically designed to handle large volumes of data. These databases provide efficient storage and retrieval mechanisms to handle the scale and complexity of big data. These databases offer features like scalability, high availability, and flexible data models to support big data use cases.

► **Examples:** Apache Cassandra, HBase, and MongoDB.

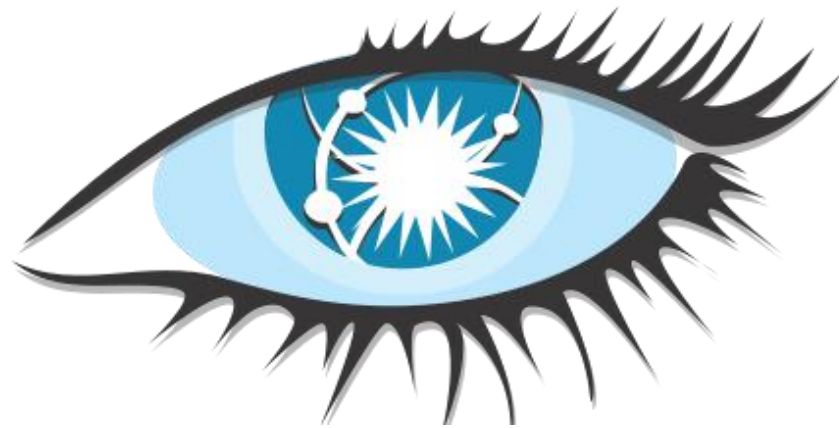
# MongoDB


It is a schema-less database that stores data in JSON-like documents and provides horizontal scaling and a distributed architecture. It has a flexible data model for accommodating unstructured and semi-structured data.



# Apache Cassandra

Highly scalable NoSQL database with a distributed and decentralized architecture for horizontal scalability and fault tolerance. It provides no single point of failure, ensuring high availability.



The background of the slide is a composite of two main visual elements. On the left, there is a vertical strip featuring a bokeh effect of out-of-focus lights in shades of blue, green, and orange, suggesting a digital or network environment. On the right, there are several overlapping, semi-transparent geometric shapes in various shades of blue and teal, creating a modern, layered aesthetic.

# Understanding the basics of message queuing and streaming with Kafka

# Agenda



Introduction to Message Queuing and Streaming



Understanding Apache Kafka



Producing and Consuming Messages with Kafka

# Introduction to Message Queuing and Streaming

## Message queuing and streaming concepts:

- Message queuing: The process of sending, storing, and receiving messages between systems or applications asynchronously.
- Streaming: The continuous flow of data in real-time from one point to another, enabling immediate processing and analysis.

## Importance of real-time data processing:

- Enables businesses to make timely decisions based on up-to-date information.
- Facilitates real-time monitoring and alerting for critical systems.
- Supports event-driven architectures and reactive systems.



# What is Apache Kafka?

- Apache Kafka is a distributed event streaming platform.
- It is designed to handle high volumes of data streams in real time.
- Kafka provides scalable, fault-tolerant, and durable messaging capabilities.
- Kafka can be used as a real-time database as it keeps the messages in logs.

<https://kafka.apache.org/>

# Key components of Apache Kafka

- ▶ Producers: Applications that publish messages to Kafka topics.
- ▶ Topics: Categories or streams of related messages.
- ▶ Consumers: Applications that subscribe to and process messages from Kafka topics.
- ▶ Brokers: the Kafka processing units

# Events in Apache Kafka

Events are published as messages to the brokers and can be composed of:

- ▶ Event key: "Alice"
- ▶ Event value: "Made a payment of \$200 to Bob"
- ▶ Event timestamp: "Jun. 25, 2020 at 2:06 p.m."




# Kafka Architecture Overview

Kafka's publish-subscribe model:

- ▶ Producers publish messages to topics.
- ▶ Consumers subscribe to topics and receive messages in real time.
- ▶ Multiple consumers can subscribe to the same topic, forming consumer groups for load balancing and scalability.

Kafka's fault-tolerance and scalability:

- ▶ Kafka replicates messages across multiple brokers for data redundancy.
  - ▶ These brokers can be added or removed dynamically to handle changing data volumes and traffic.
  - ▶ Kafka supports horizontal scalability by distributing partitions across brokers.
- 



# Kafka Architecture Overview

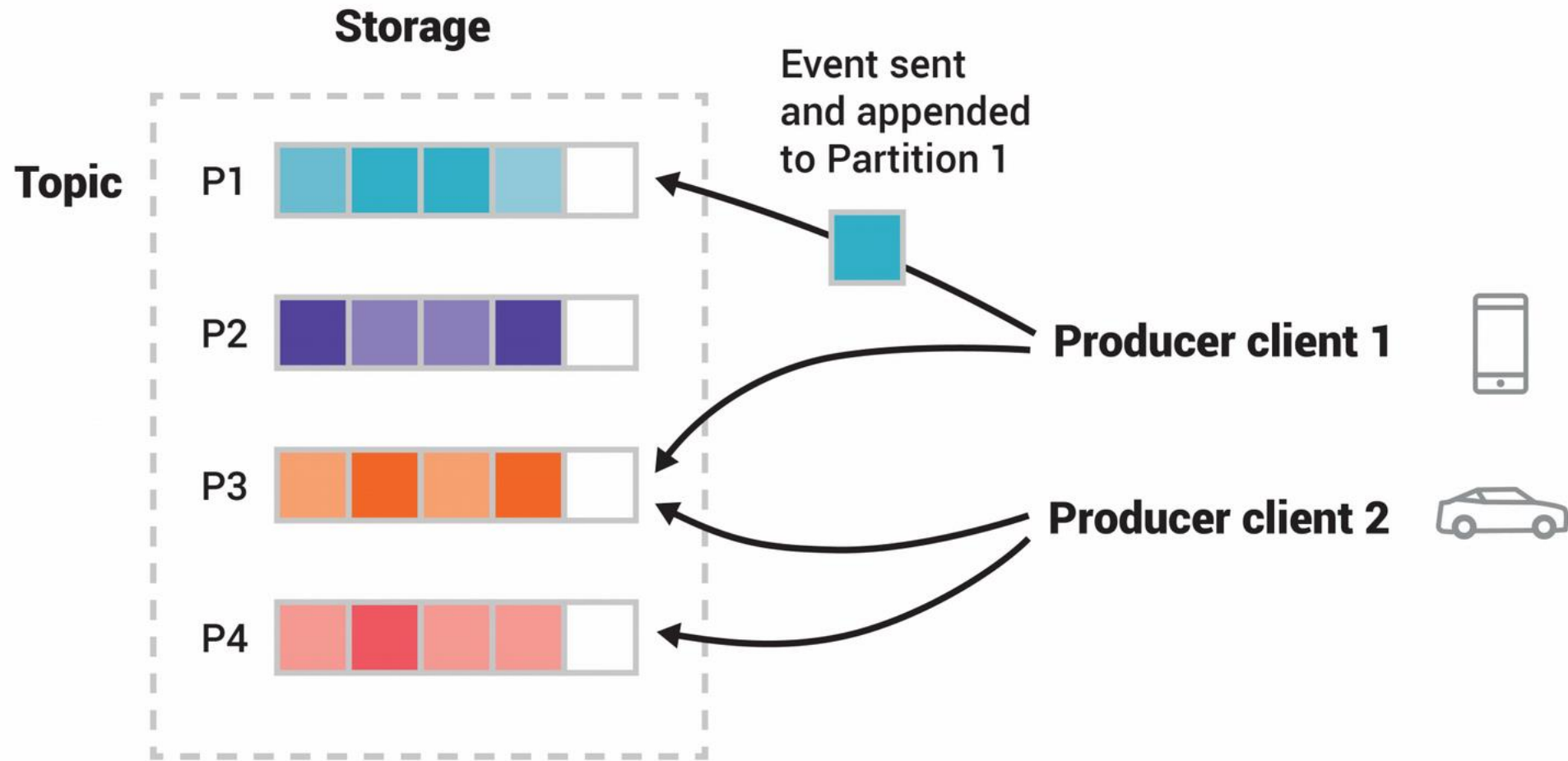
Kafka's durability and persistence:

- ▶ Messages written to Kafka topics are persisted to disk, ensuring data durability.
- ▶ Kafka retains messages for a configurable period, enabling data replay and batch processing.
- ▶ Disk-based storage allows large amounts of data to be stored for extended periods.

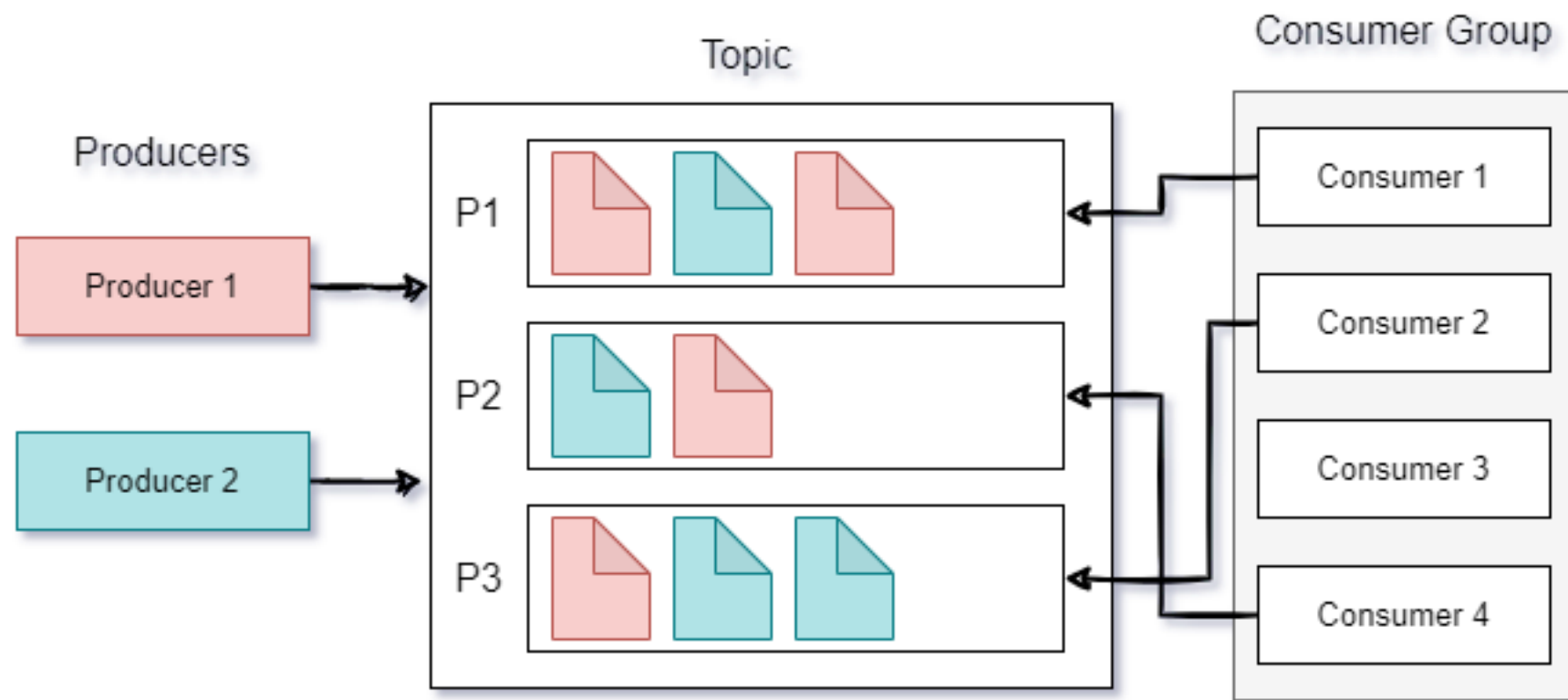
Kafka's high throughput and low latency:

- ▶ Kafka can handle high message volumes per second.
- ▶ It achieves low latency by efficiently buffering and batching messages.
- ▶ Kafka's design minimizes disk I/O and optimizes network transfer for fast data processing.

# How Kafka looks like



# How Kafka looks like



# Understanding partitions and their benefits

- ▶ Partitions are the basic unit of data organization in Kafka.
- ▶ Each topic is divided into one or more partitions.
- ▶ Partitions enable parallelism and scalability by allowing multiple consumers to read from different partitions concurrently.
- ▶ Partitioning also helps distribute data across multiple brokers in a Kafka cluster.





## Consumer groups and their role in parallel processing:

- Consumer groups enable parallel processing of messages in Kafka.
- Consumer group consists of multiple consumers that collectively consume messages from a topic.
- Each consumer in a group is assigned a subset of partitions from the subscribed topic.
- Consumer groups allow for load balancing and scalability by distributing the message processing workload among consumers.

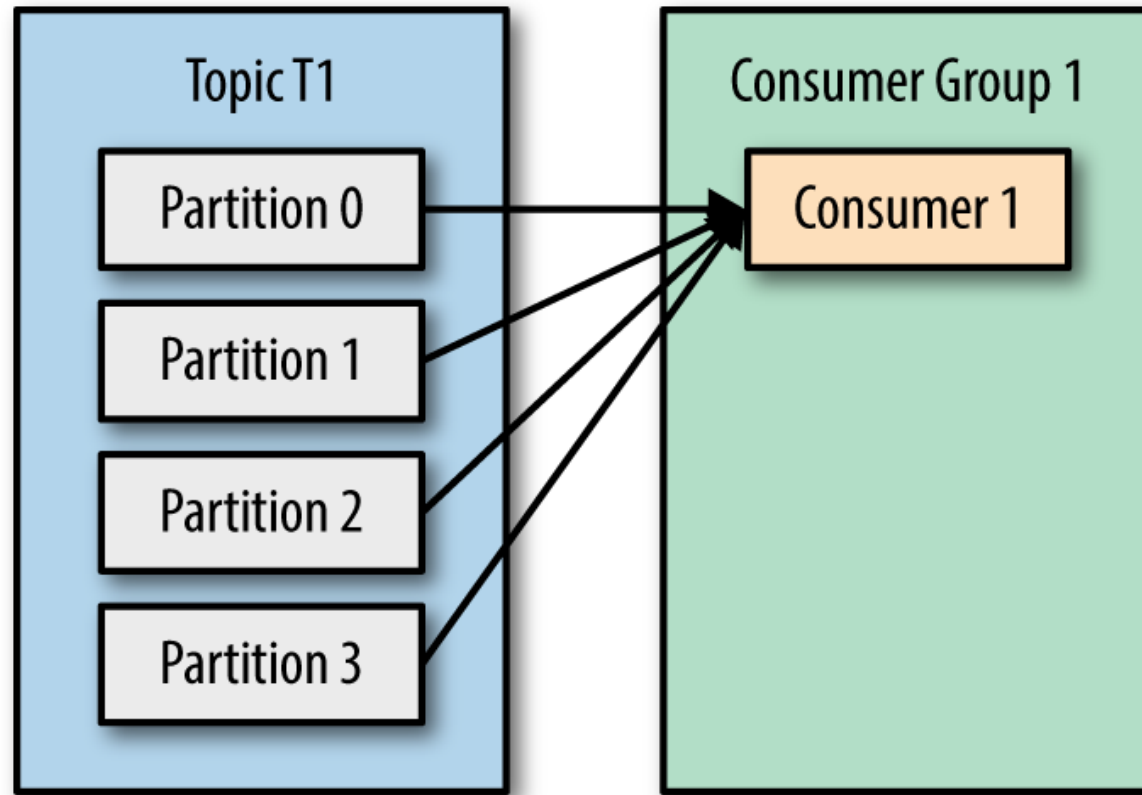
# Exactly-once policy

- Exactly-once policy ensures that each message is processed only once by each consumer group, guaranteeing data integrity within the consumer group.
- It means that a message is either successfully processed and committed or not processed at all within the group

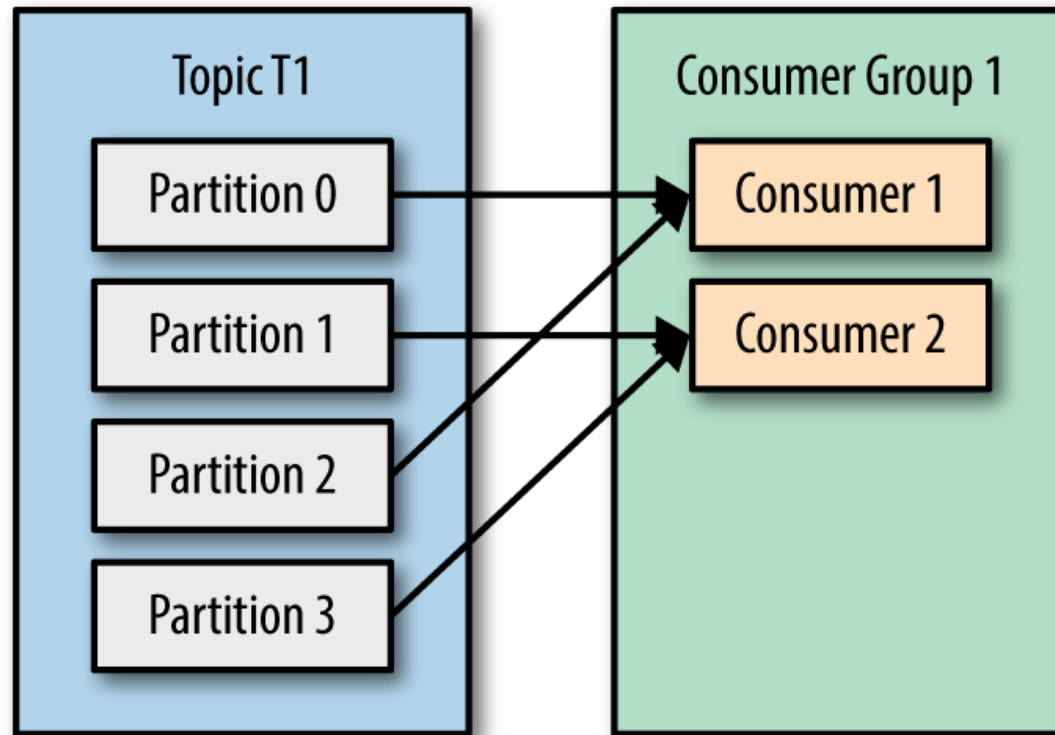
# Order of reading messages

- ▶ Messages are read from the topics using the policy of first-in first-out.
- ▶ Messages are split evenly among topic partitions.
- ▶ If there are more partitions than consumers, messages are read from the partitions evenly.
- ▶ If messages must be read in the exact order they have been produced → Only one partition must be used, otherwise, the order is not guaranteed.

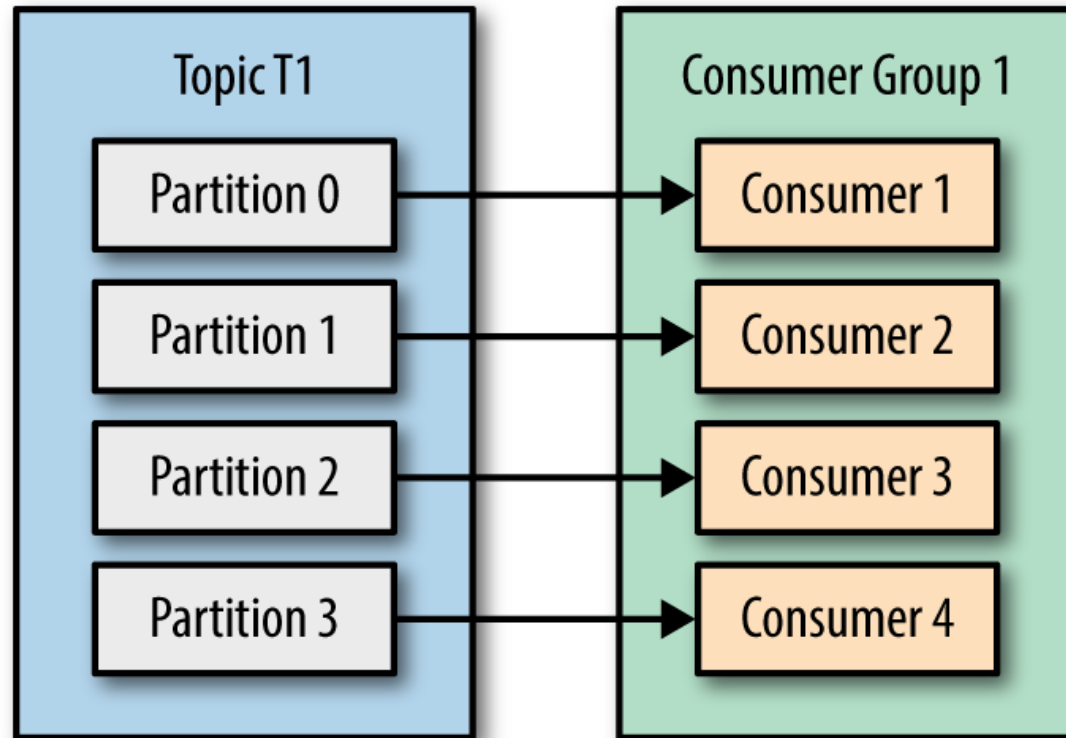
# Consumer Groups



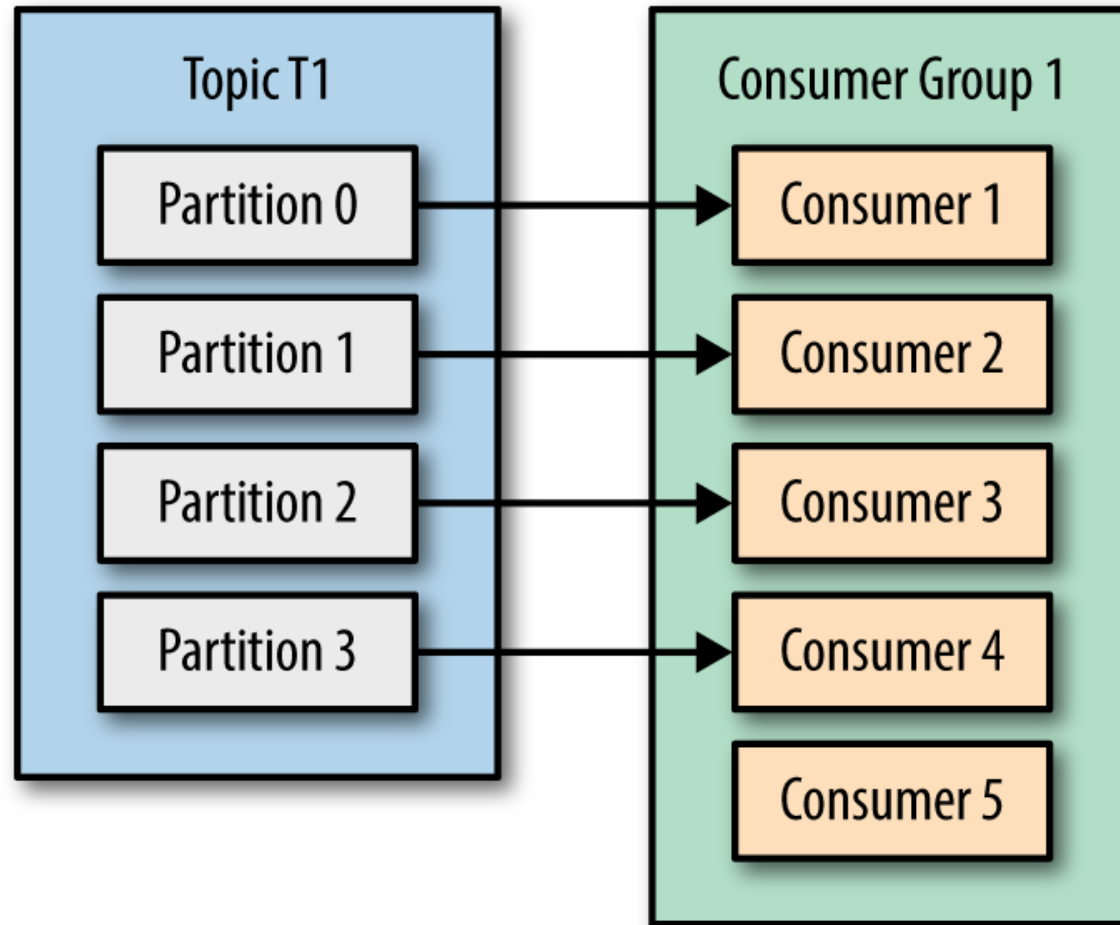
# Consumer Groups



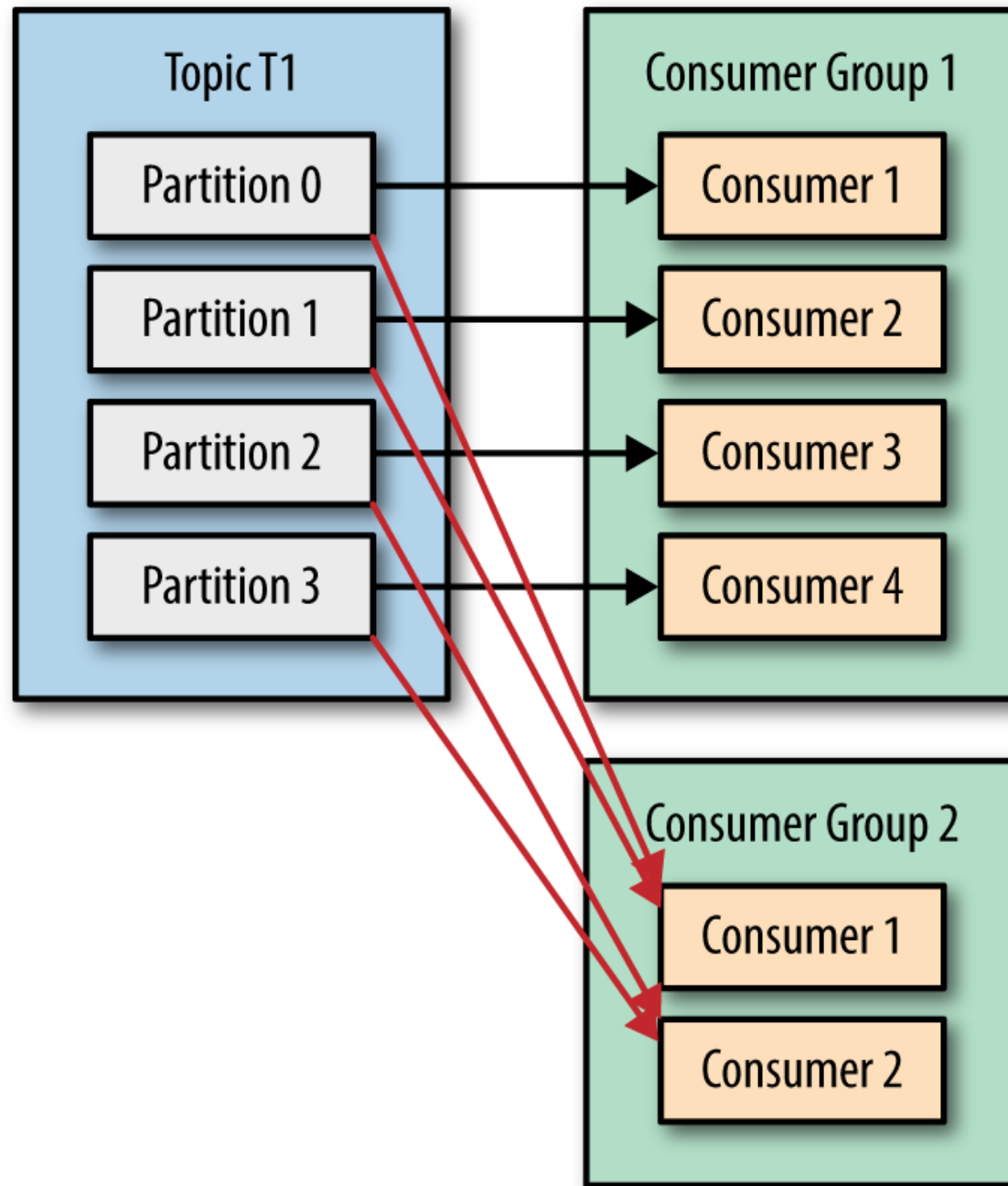
# Consumer Groups



# Consumer Groups



# Consumer Groups





# Replication for fault tolerance and high availability

- ▶ Kafka uses replication to ensure fault tolerance and high availability of data.
- ▶ Each partition in Kafka can have multiple replicas, where each replica is stored on a different broker.
- ▶ Replication ensures that data is not lost in case of broker failures.
- ▶ Replicas also provide load balancing and allow for seamless failover.

# Creating Topics and Producing Messages

- Topics can be created using the Kafka command-line tools or programmatically through Kafka APIs.
- When creating a topic, you specify the topic name, the number of partitions, and the replication factor.
- Topics can be configured with various options such as retention and clean-up policies.
- Retention policy determines how long Kafka retains the messages in a topic.
- Clean-up policy defines the criteria for removing or compacting old messages from a topic.

# Producing Messages

- ▶ Producers are responsible for publishing messages to Kafka topics.
- ▶ Producers can be implemented using Kafka client libraries in various programming languages.
- ▶ To produce messages, you need to specify the topic to which the messages should be published.
- ▶ Producers can also specify additional parameters like message key, partitioning strategy, and message serialization format.

# Kafka Consumers

- ▶ Consumers subscribe to one or more topics and consume messages published on those topics.
- ▶ Kafka consumers can be implemented using Kafka client libraries in different programming languages.

# ZooKeeper coordination and its role in Kafka:

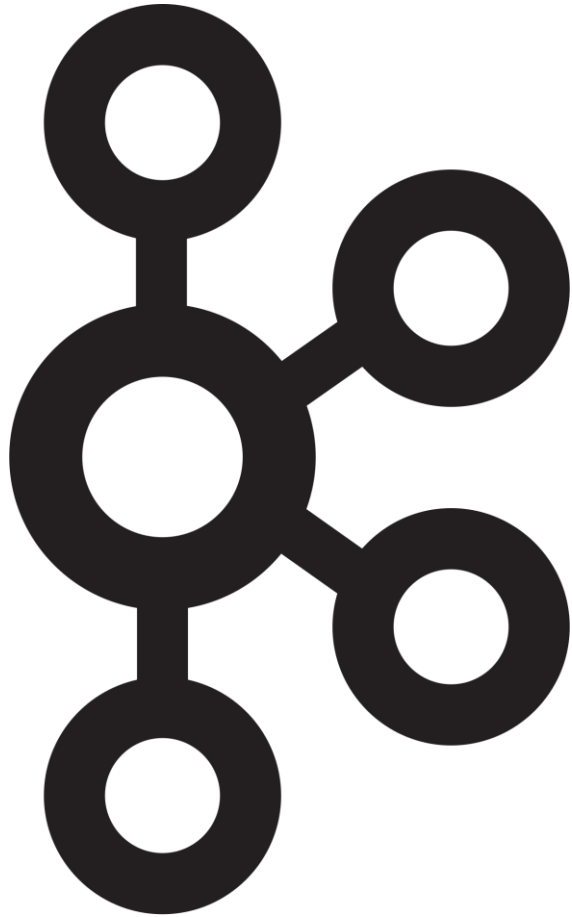
- ▶ ZooKeeper is a distributed coordination service used by Kafka for maintaining cluster metadata.
- ▶ It stores information about Kafka topics, partitions, and brokers.
- ▶ ZooKeeper provides leader election and failover capabilities for Kafka brokers.
- ▶ Kafka brokers communicate with ZooKeeper to register themselves, handle reassignments, and track cluster state changes.
- ▶ ZooKeeper ensures the consistency and synchronization of the Kafka cluster.

# From January 2024 Zookeeper will be deprecated in Kafka

- ▶ On August 2022, Apache Kafka announced a new protocol (Kraft) to substitute Zookeeper. Among its benefits, it reduces the need of maintaining two technologies and substantially improve the scalability of Kafka.
- ▶ On January 2024, Zookeeper is planned to be removed.

An example of the new configuration for Kraft using docker:

<https://github.com/confluentinc/cp-all-in-one/blob/master/cp-all-in-one-kraft/docker-compose.yml>



# Hands-on tutorial on Apache Kafka and its applications in handling big data for AI

# Scaling Kafka for Large Workloads

Techniques for scaling Kafka to handle large volumes of data:

- Horizontal scaling by adding more Kafka brokers to distribute the data across multiple nodes.
- Partitioning data across multiple partitions to enable parallel processing and increase throughput.
- Increasing the number of consumer instances to handle higher message consumption rates.



# Setting up a cluster

- ▶ Horizontal scalability is achieved by adding more Kafka nodes (Kafka brokers) to form a cluster.
- ▶ Nodes must have a different ID
- ▶ Each node typically defines three ports to handle different types of connections: (1) external clients, (2) connections from the same machine, (3) connections from the same network. Further reading on handling ports and connections:  
<https://docs.confluent.io/platform/current/kafka/multi-node.html>

# Using the REST API service for Apache Kafka

- ▶ Apache Kafka offers an additional REST API service tool
- ▶ The REST API is installed separately from Apache Kafka
- ▶ It allows to produce and consume messages using a REST API
- ▶ Clients only to connect to the REST API to use Kafka
- ▶ It removes the need of deploying producers and consumers in every application.

# Kafka Security Best Practices

Authentication and authorization mechanisms in Kafka:


- Authentication: Implementing mechanisms such as SSL/TLS certificates or SASL (Simple Authentication and Security Layer) for verifying the identity of clients connecting to Kafka.
- Authorization: Configuring fine-grained access controls to determine which clients have permission to perform specific operations on Kafka topics and resources.





# Monitoring and Managing Kafka

Best practices for managing and maintaining Kafka clusters:

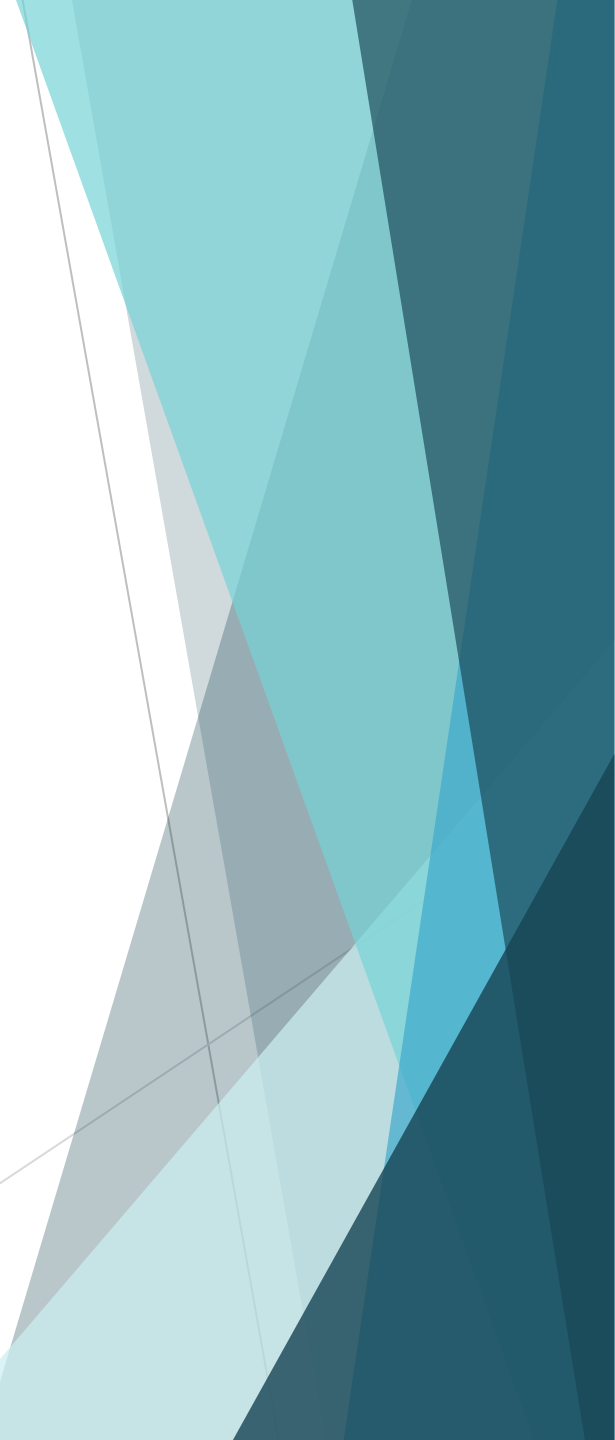
- Regularly monitor cluster health, including brokers, topics, partitions, and replication status.
  - Ensure sufficient hardware resources for Kafka brokers, such as CPU, memory, and disk space.
  - Implement proper data retention policies and perform regular backups to prevent data loss.
  - Stay updated with the latest Kafka versions and apply patches and security updates.
  - Implement access controls and security measures to protect sensitive data and prevent unauthorized access.
- 





# Monitoring and Managing Kafka

Monitoring tools and metrics for Kafka:

- Various monitoring tools are available, such as Confluent Control Center, Prometheus, Grafana, and Apache Kafka Manager.
  - Key metrics to monitor include throughput, latency, broker and topic-level metrics, consumer lag, and resource utilization.
- 

# Kafka Troubleshooting

Troubleshooting techniques and best practices:

- Log analysis: Reviewing Kafka logs to identify errors, warnings, and performance issues.
- Monitoring metrics: Monitoring key metrics such as throughput, latency, and disk utilization.
- Partition rebalancing: Resolving uneven data distribution across partitions by rebalancing consumers.
- Network and connectivity checks: Verifying network configurations, firewalls, and connectivity to brokers.
- Configuration review: Examining Kafka configuration parameters to ensure optimal settings.

# Kafka Integration

- ▶ Apache Kafka and the open-source community have developed many frameworks to connect Kafka with Big Data and AI tools.
- ▶ Examples are the integration with Apache Spark and TensorFlow.

# Kafka Integration with Apache Spark

Integration with Apache Spark for data analytics:

- Kafka and Spark can work together to process streaming real-time data, but also batch data.
- Spark Streaming allows for high-throughput, fault-tolerant real-time stream processing.
- Kafka's direct integration with Spark enables easy data ingestion and processing.



# Kafka and TensorFlow for AI Applications

Introduction to TensorFlow for machine learning and AI:

- TensorFlow is an open-source machine learning framework developed by Google.
- It provides a flexible and comprehensive ecosystem for building and deploying machine learning models.
- Kafka can be used as a data pipeline for feeding training data to TensorFlow models.
- Streaming data from Kafka can be consumed and processed by TensorFlow for real-time predictions.
- Kafka's scalability and fault tolerance ensure reliable data delivery to TensorFlow workflows.

