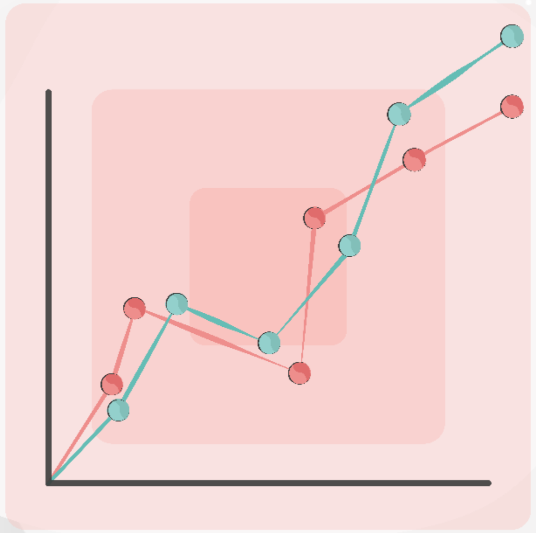
BPP

Data engineering

Module: Data pipelines

Topic: From data pipelines to data architectures

Welcome to today’s webinar.



Ice breaker

Building Careers Through Education

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

Scale it or fail it!

"Imagine you're hosting a dinner party for 4 people. Suddenly, 40 guests show up. What's your first move?"

Your options:

A. Order 10 pizzas and hope for the best

B. Start cooking in batches and delegate tasks

C. Set up a buffet line and let people serve themselves

D. Panic and pretend it was a prank

Submit your responses to the chat or turn on your microphone

BPP

Real-world case study

Zalando's scalable data platform on AWS

Challenge

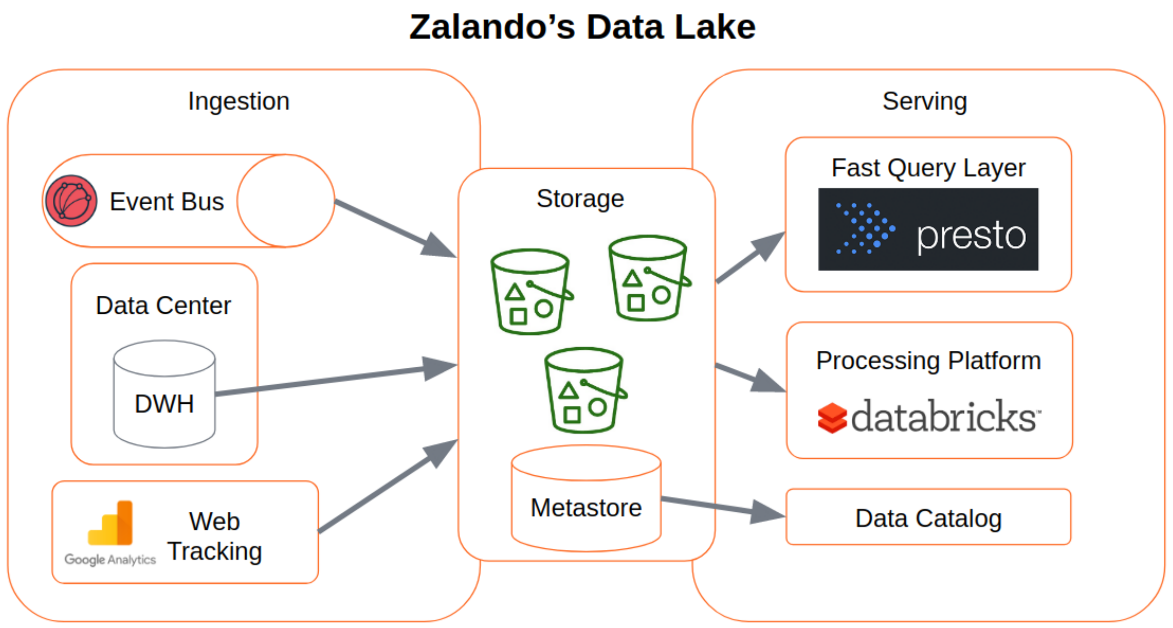
- Billions of daily events from web, mobile, and logistics
- Real-time personalization and inventory updates
- Elastic scaling for peak events like Black Friday

Solution

- Kafka + Kinesis for real-time streaming
- Serverless compute with AWS Lambda
- Modular microservices and Airflow orchestration
- Scalable storage with S3 and Redshift

zalando

Zalando delivers real-time recommendations, accurate inventory, rapid feature rollouts, and seamless performance during traffic spikes.



Zalando's Real-Time Retail Engine: *Scaling with AWS,* ***Image source:*** *AWS*

e-learning recap

Building Careers Through Education

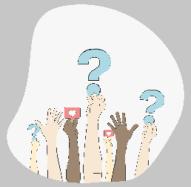
Everybody Matters. Stronger Together. Student, Learner and Client Centric. Trust and Respect. Embrace Change.

Reflecting on your learning…

The e-learning for this topic, covered the following areas:

1. Scalability means designing systems that grow with data and demand
2. Pipelines scale through horizontal scaling, stream processing, and decoupling
3. Performance and reliability require monitoring, fault tolerance, and testing
4. Scalable architectures are modular, loosely coupled, and event-driven
5. Tools like Spark, Kafka, and Airflow enable distributed, resilient workflows
6. Smart trade-offs balance speed, cost, and complexity

Q&A discussion



- Do you have any questions about any of these areas?
- Did everything in the e-learning make sense?

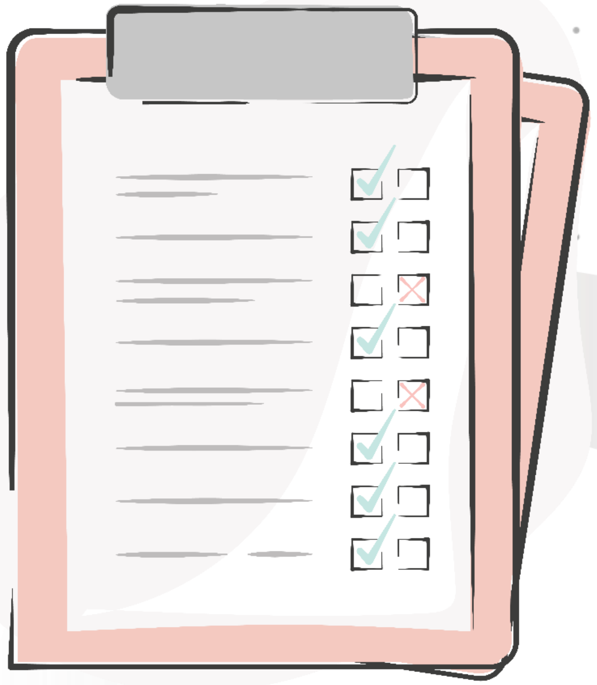
BPP

Webinar Agenda

Today, we will cover the following:

1. Core concept recap
2. Practical lab
3. Summary
4. Q&A

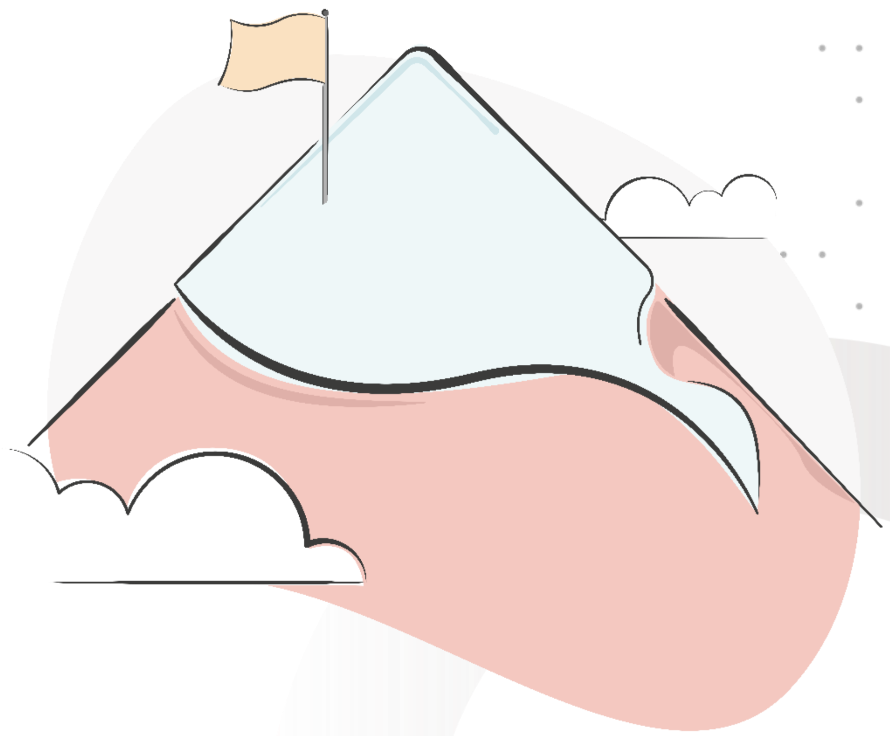
Building Careers Through Education
Everybody Matters.
Stronger Together.
Student, Learner and Client Centric.
Trust and Respect.
Embrace Change.



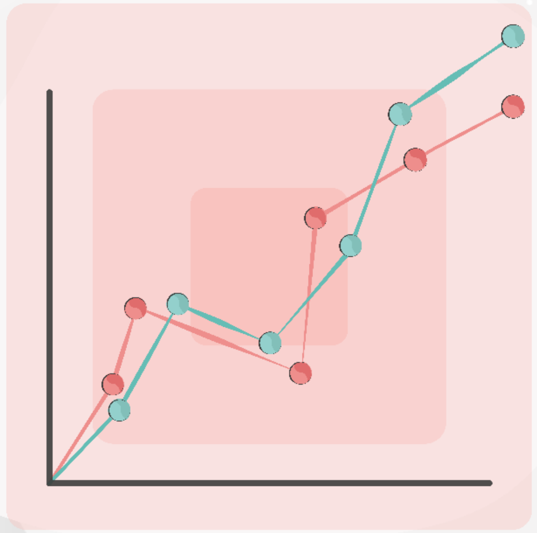
Session aim and objectives

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

- Understand what scalability means in data pipelines and architectures
- Explore strategies and tools for building scalable systems
- Learn from real-world examples of scalable data design



Recap of core concepts



What is scalability?

From prototype to production

- Scalability is the ability of a system to grow with demand
- It applies to both data pipelines and overall architecture
- Critical for performance, reliability, and cost-efficiency

Scalability is the ability of your data solution to grow as demand increases.

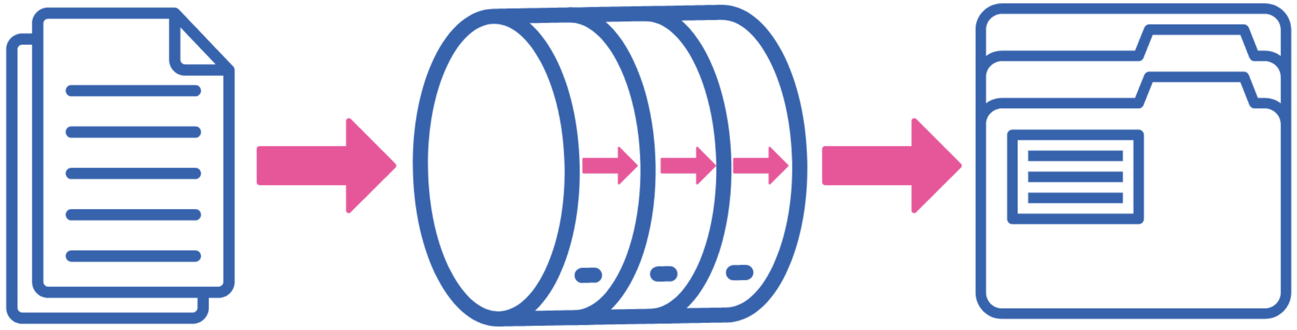


Figure: *An abstraction of a data pipeline*

Scaling data pipelines

Strategies for growth

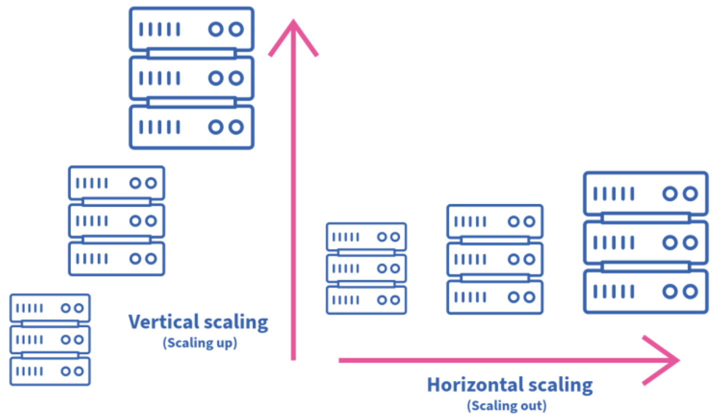


Figure: Vertical vs. Horizontal Scaling

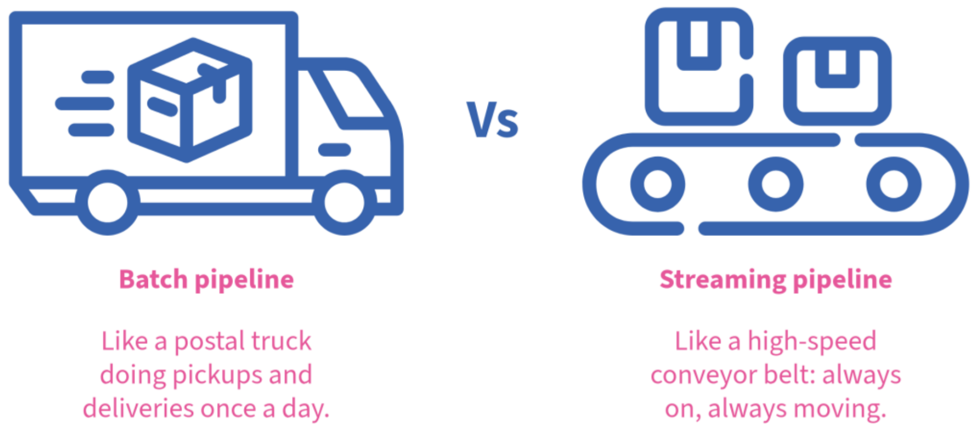

Vs
Batch pipeline
Like a postal truck doing pickups and deliveries once a day.
Streaming pipeline
Like a high-speed conveyor belt: always on, always moving.

Figure: Difference between Batch and Streaming via an analogy.

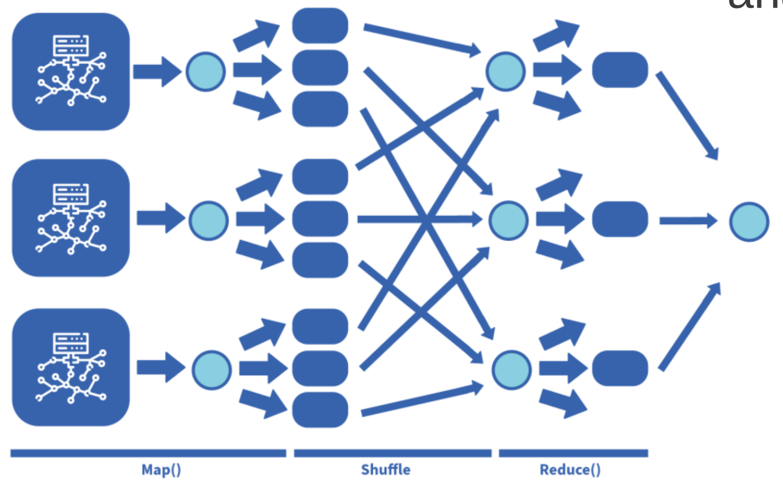


Figure: Diagram of how MapReduce works

Tools for scalable pipelines

Building Careers Through Education

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

The Engineer’s Toolkit

Apache Spark: distributed processing for large-scale data

Apache Kafka: real-time messaging and decoupling

Airflow & Prefect: orchestrate complex workflows

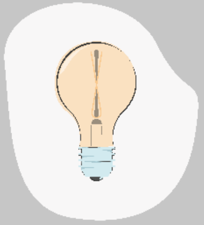
Ensuring performance

Monitoring and optimisation

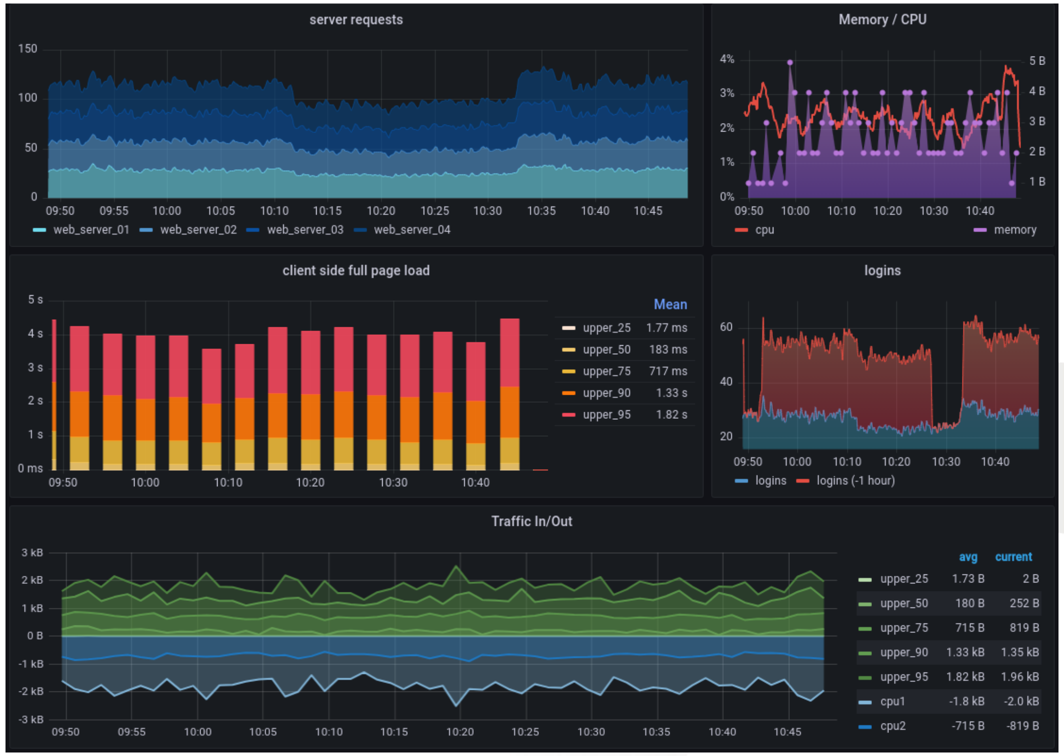
Building Careers Through Education

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

- Track throughput, latency, and resource usage
- Optimise transformations to reduce compute load
- Choose storage formats that match your workload



Scalable systems depend on visibility - monitoring and optimising performance is essential to ensure your pipelines stay fast, efficient, and reliable as they grow.



Ensuring Performance: Monitoring and optimisation strategies for scalable data pipelines, Image source: Grafana.com

BPP

Building for reliability

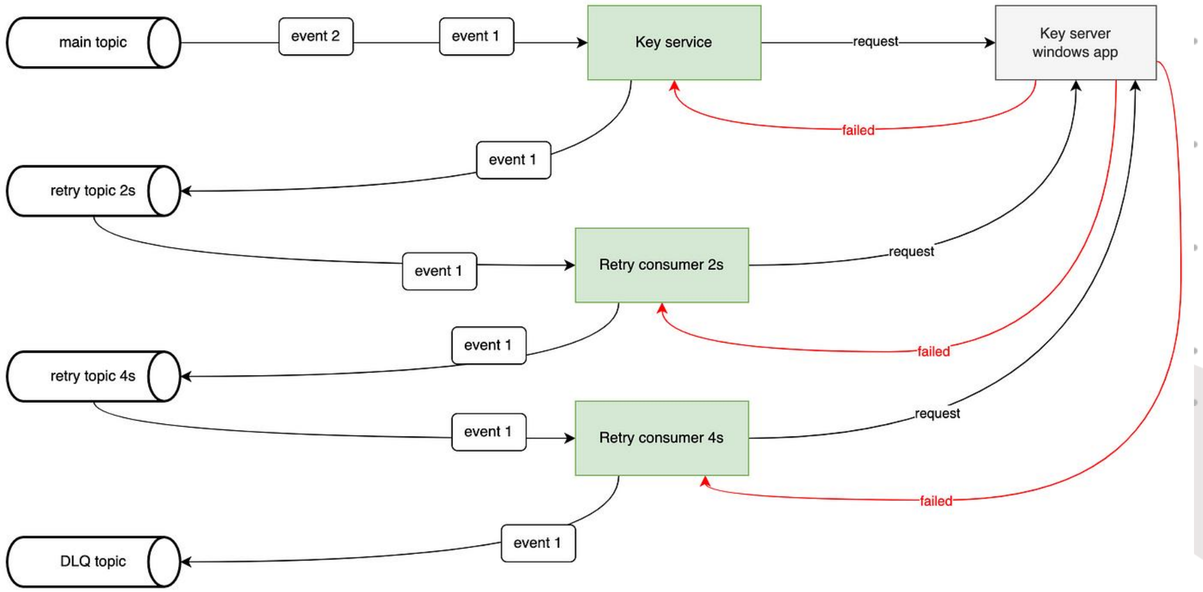
Building Careers Through Education

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

Fault tolerance and resilience

- Use retries, checkpoints, and idempotent design
- Decouple components with queues like Kafka
- Validate data early and test beyond unit tests

Reliability isn't about avoiding failure - it's about designing systems that recover gracefully, validate early, and keep data trustworthy at scale.



Resilient Event Processing: How Retry Logic and Dead Letter Queues Keep Pipelines Reliable, image source: Medium.com

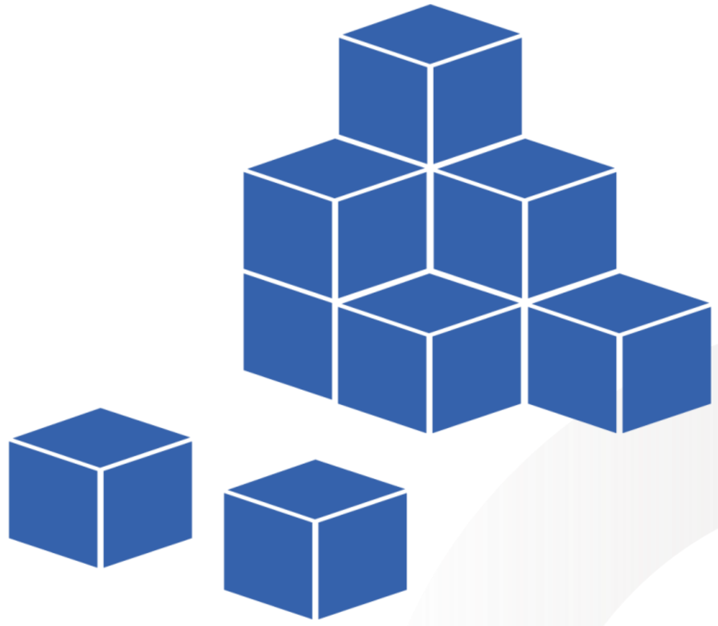
BPP

Designing scalable architectures

Thinking beyond pipelines

- Modular, loosely coupled components scale better
- Event-driven design enables real-time responsiveness
- Use the right storage for the right data type

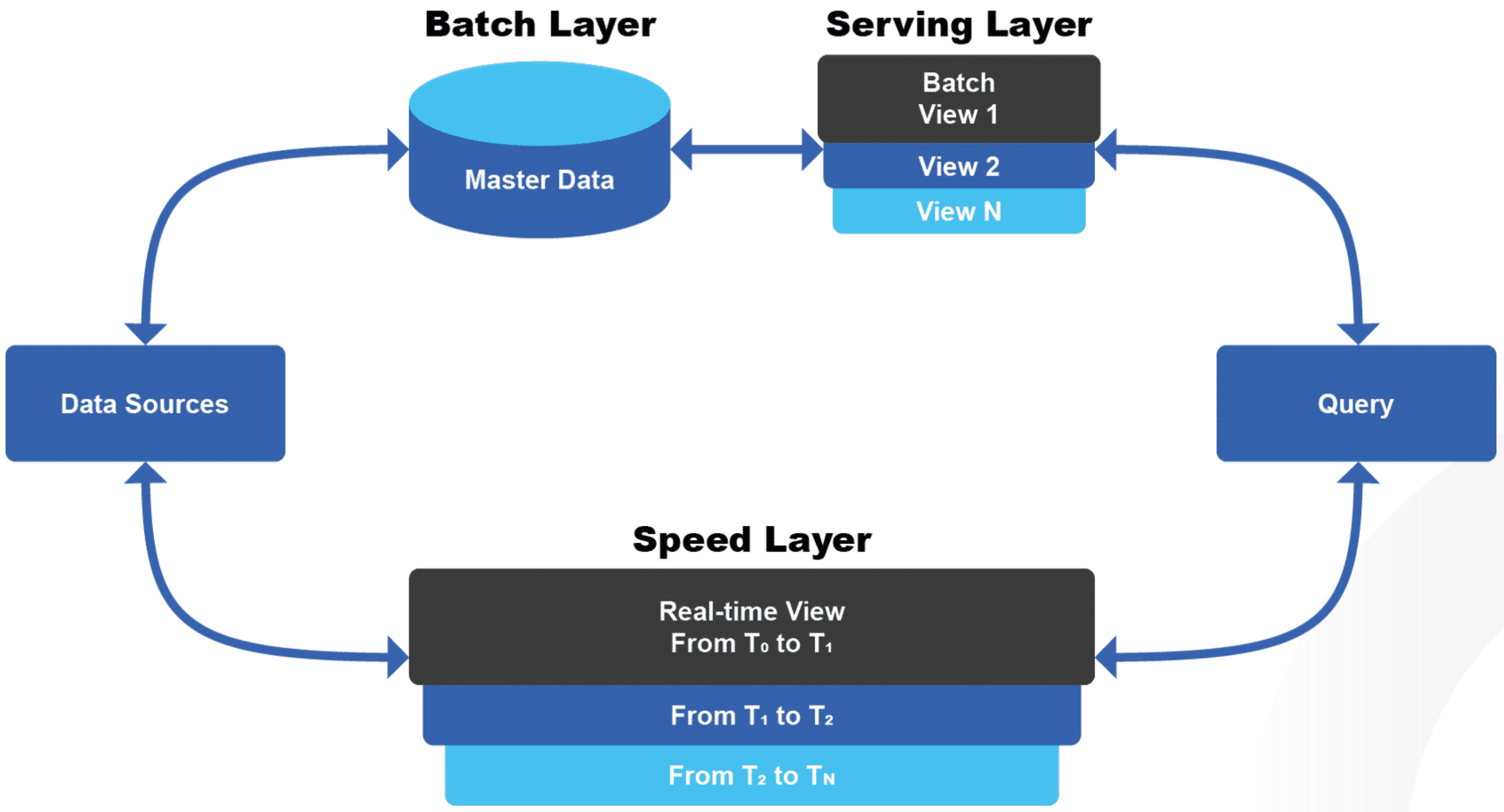
Scalable architectures are built, not bolted on - modularity, event-driven design, and smart storage choices are the foundation for systems that grow and evolve.



Scalable by Design: Modular, Architectures That Grow with Your Data, image source: freepic.com

Architecture patterns

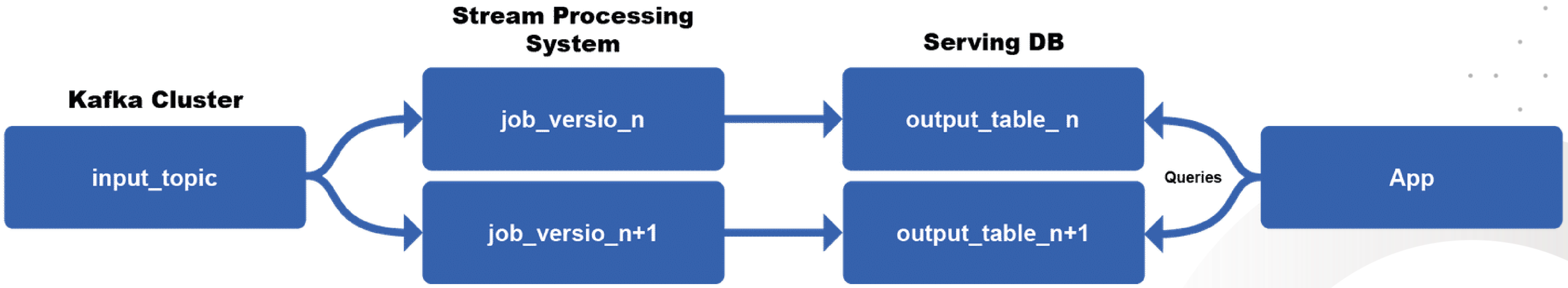
Lambda



Lambda Architecture combining batching and streaming

Architecture patterns

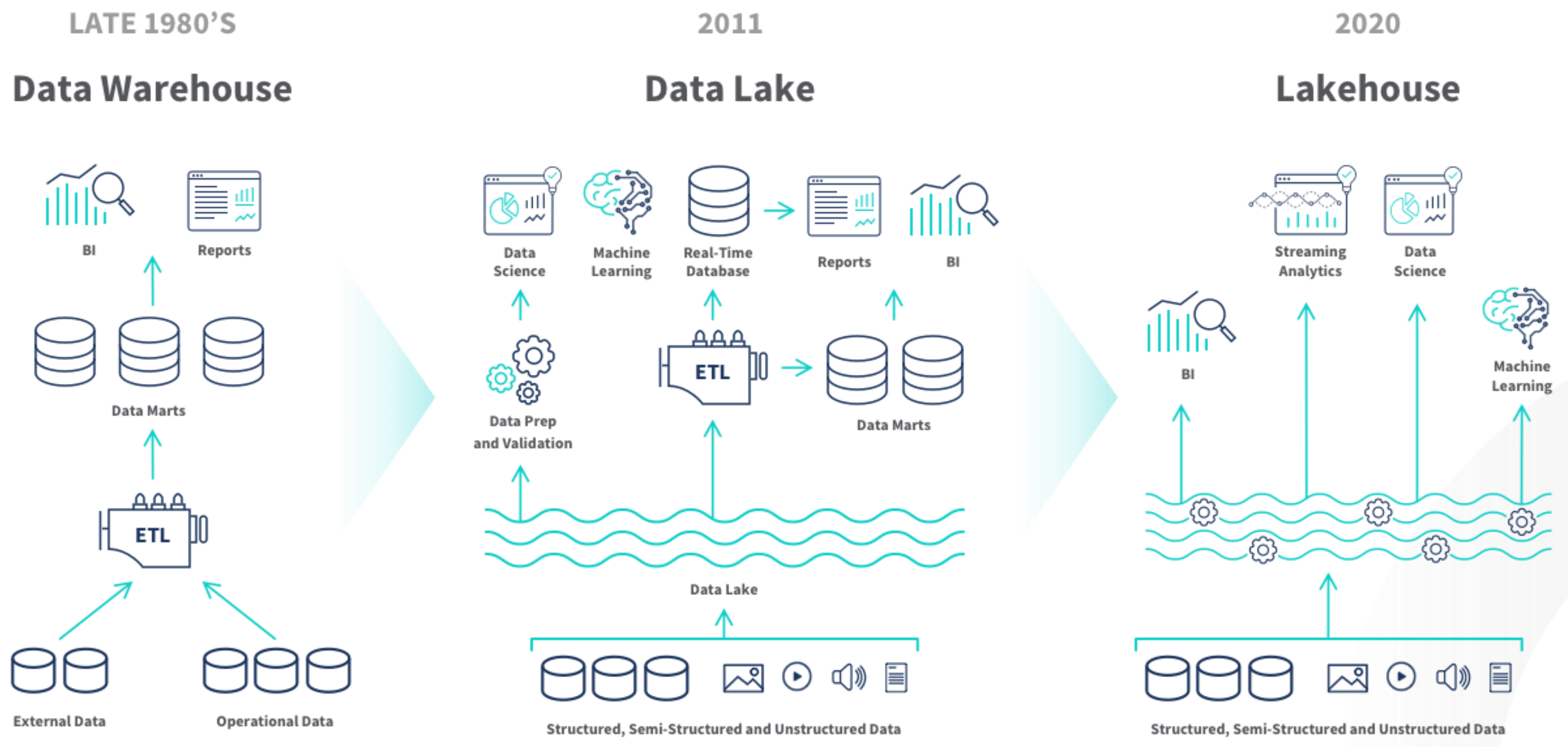
Kappa



Kappa Architecture for streaming based process

Architecture patterns

Lakehouse



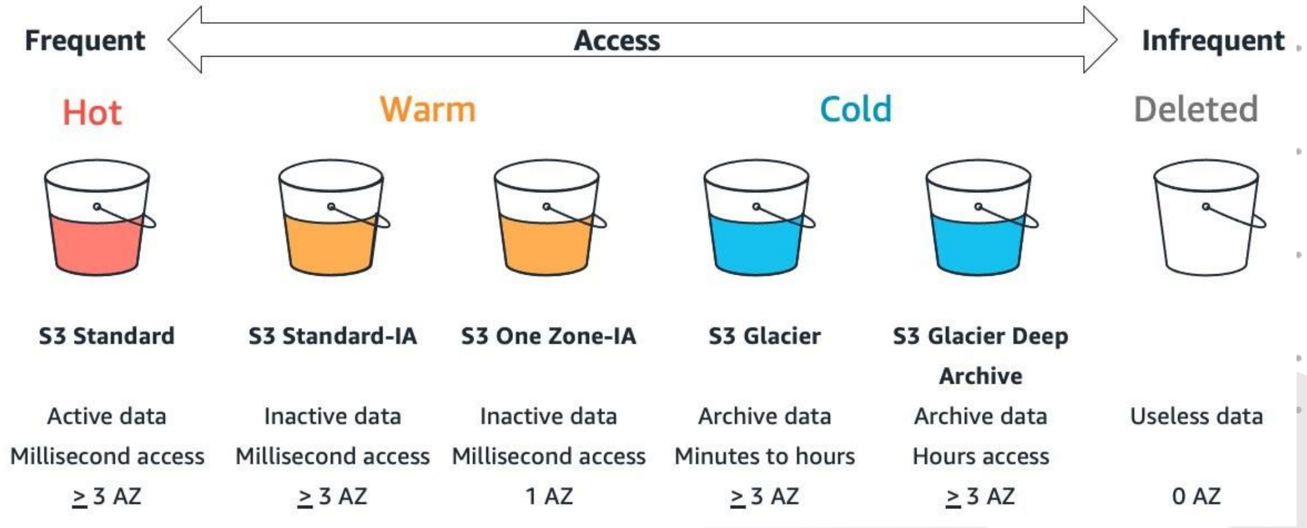
Kappa Architecture, which is a purely streaming based process, image source: HAZELCAST

Balancing cost and performance

Smart scaling

- Store cold data cheaply; query hot data efficiently
- Use batch jobs to reduce always-on compute costs
- Auto-scale with guardrails to avoid surprise bills

Scalability isn't just about speed - it's about making smart, cost-effective choices that align with your workload and business goals.



Smart Scaling in the Cloud: Matching Storage and Compute to Your Workload Needs

Real – World case studies

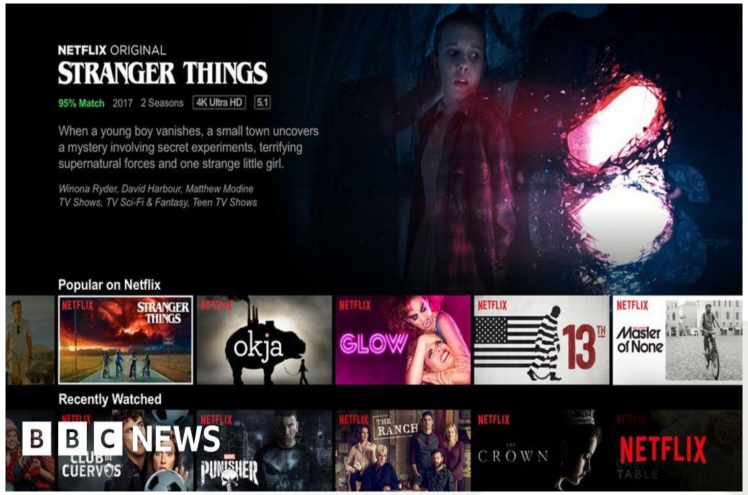
Lessons from the field…

Ride-sharing: Kafka + Flink for real-time GPS data

Image source: ottocar.co.uk

Retail: Lakehouse model for unified analytics

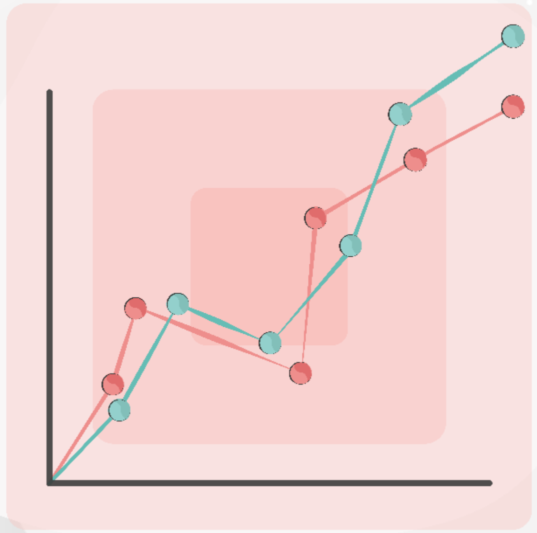
Image source: Webandcrafts.com


NETFLIX ORIGINAL
STRANGER THINGS
95% Match 2017 2 Seasons 4K Ultra HD 5.1
When a young boy vanishes, a small town uncovers a mystery involving secret experiments, terrifying supernatural forces and one strange little girl.
Winona Ryder, David Harbour, Matthew Modine
TV Shows, TV Sci-Fi & Fantasy, Teen TV Shows
Popular on Netflix
Recently Watched

Media: Auto-scaling clusters for burst traffic

Image source: BBC.co.uk

Practical lab



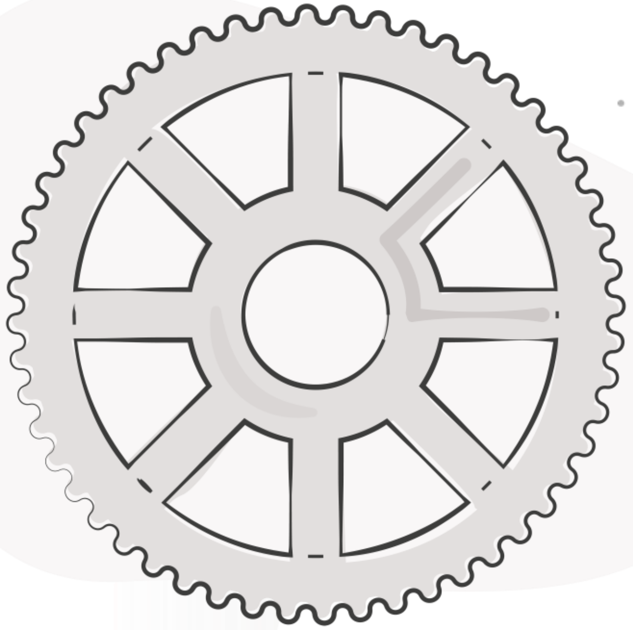
Exercise part 4

Next phase of design based on Scottish data

- **Update the Database Schema:** Update your database schema to accommodate data from the Scottish version of the software application.
- **Re-implement the Data Pipeline:** Re-implement the pipeline to clean, validate, and load the four data sets into your revised database schema.
- **Utilise Additional Test Data:** Use additional test data provided by the Scottish application team, including data for ten Scottish users and the login timestamps.
- **Stakeholder Support:** Your instructor will act as the stakeholder SM and answer any questions about the new data.
- **Documentation Requirements:** Create full documentation for both th schema and the pipeline for implementation by other teams.

Files Provided in the Hub:

- The 10 sample records are in the "SC User Data.csv" file
- The login audit file is "SC-User-LoginTS.csv."


Practical challenge

Key Learning Summary

- **Scalability Fundamentals -** Systems must handle growing data volume, complexity, and user demand without performance loss—across both pipelines and architectures.
- **Scaling Data Pipelines -** Use horizontal scaling, stream processing, partitioning, and decoupling (e.g., Kafka) to improve throughput and resilience.
- **Key Tools for Scale -** Leverage Apache Spark for distributed processing, Kafka for real-time streaming, and Airflow/Prefect for orchestration.
- **Performance & Reliability -** Monitor metrics like latency and resource usage, optimise transformations, and build in retries, validation, and fault tolerance.
- **Architectural Design Principles -** Design modular, loosely coupled systems using event-driven flows and polyglot persistence for flexibility and maintainability.
- **Architecture Patterns -** Choose from Lambda (batch + stream), Kappa (stream-only), or Lakehouse (unified analytics) based on your data needs.
- **Cost-Performance Trade-offs -** Store cold data cheaply, use batch jobs to reduce compute costs, and auto-scale with guardrails to avoid surprise bills.

Any questions or feedback?

Building Careers Through Education

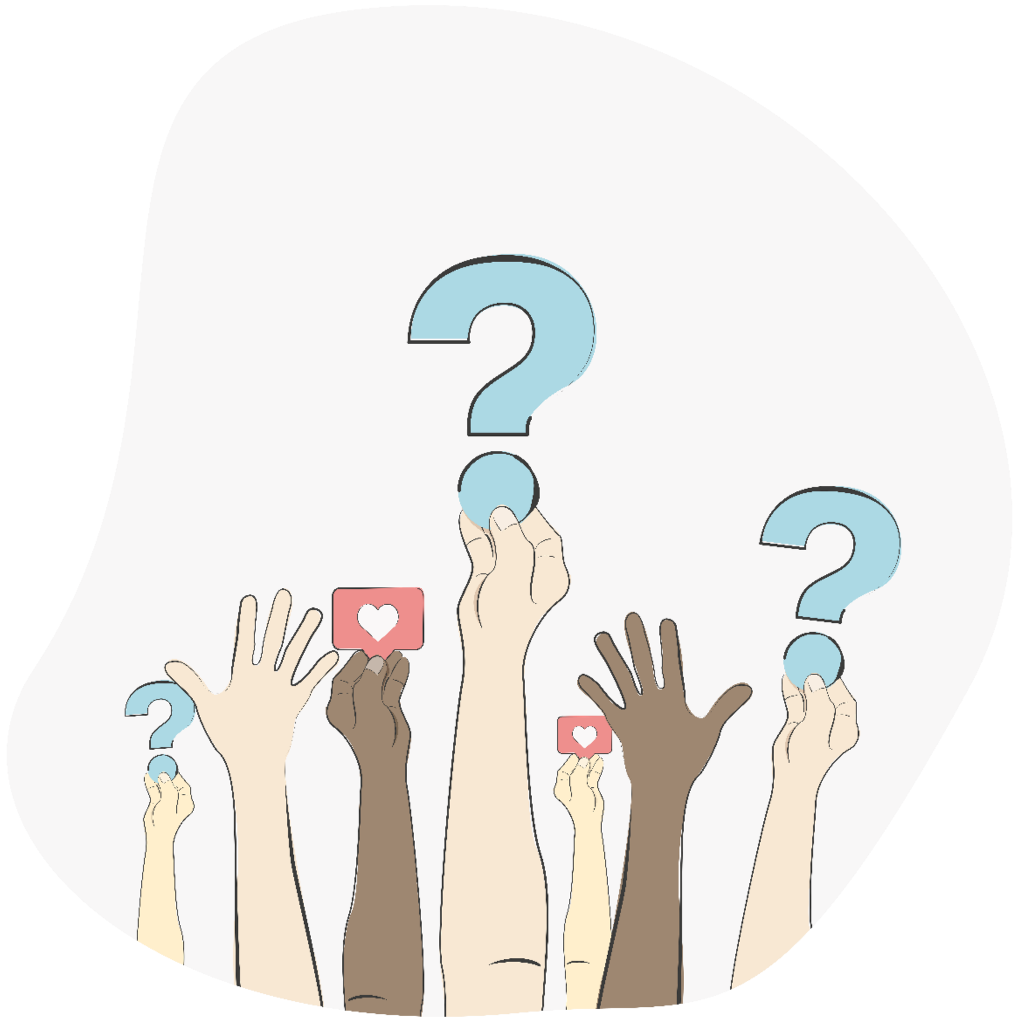
Everybody Matters.

Stronger Together.

Student, Learner and Client Centric.

Trust and Respect.

Embrace Change.



BPP

Thank you