Stream Processing with Apache Kafka & .NET

Niels Berglund - Architect Lead Derivco, Data Platform MVP
niels.it.berglund@gmail.com
https://nielsberglund.com
https://twitter.com/nielsberglund

Agenda

- Kafka 101
- Kafka & .NET
- Stream processing
- ksqlDB

Data

*Civilization has always run on data **

*Today, every aspect of human life is fueled by data **

*Data has become a renewable resource **

*Even when trade slows, data continuous to grow at a steady pace **

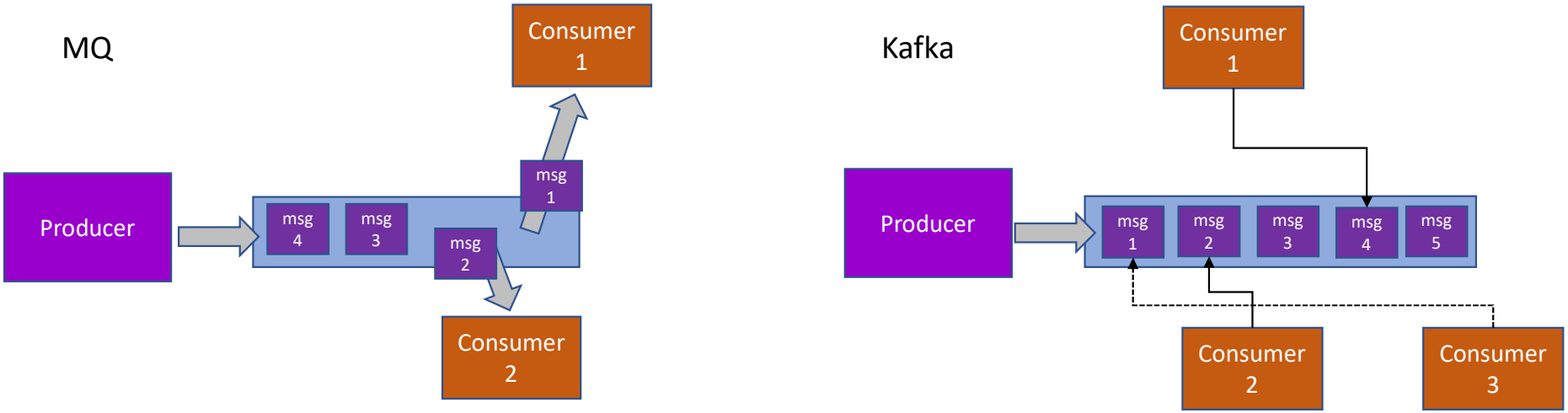
- 2016 - 16 Zettabytes of data
- 2025 - 165 Zettabytes of data

*** Brad Smith: Tools and Weapons - The Promise and the Peril of the Digital Age**

Kafka 101 - What is Kafka?

- Started at LinkedIn to replace their messaging systems.
  - Open sourced 2014
  - Confluent the commercial arm of Kafka
- Distributed streaming platform.
  - Publish and consume streams of records (events).
  - Persist published records.
  - Process streams of records in real-time.
- Commit log.
  - Logs are trustworthy.
  - Distributed for scale and resilience.

Kafka vs. Message Queues

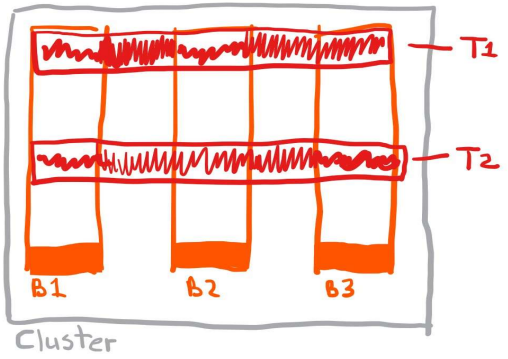


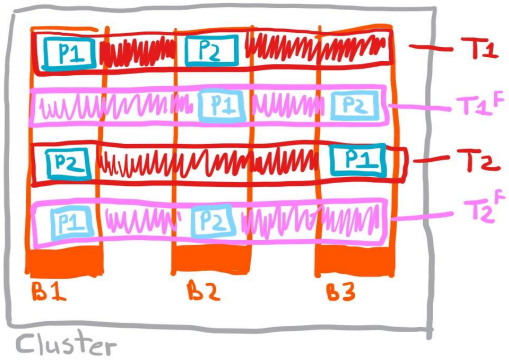
Kafka 101 - Key Concepts

- Cluster
- Broker
- Topic
- Partition
- Offset
- Consumer groups

Kafka 101 - Topics & Partitions

- Topics
  - Name/definition of a stream of msgs/records/events, (wager, login, order, etc.).
  - Schema (JSON, Avro, CSV, Protobuf).
  - A message passed in to a topic has: key, value, timestamp.
  - Properties of a topic: retention period, compression type, number of partitions, replication factor.
- Partitions
  - A topic has partitions, (1 - n).
  - A partition is the actual log.
  - Partition provides scalability, parallelism, order.
  - Publisher controls the partitioning, (the key in the message).

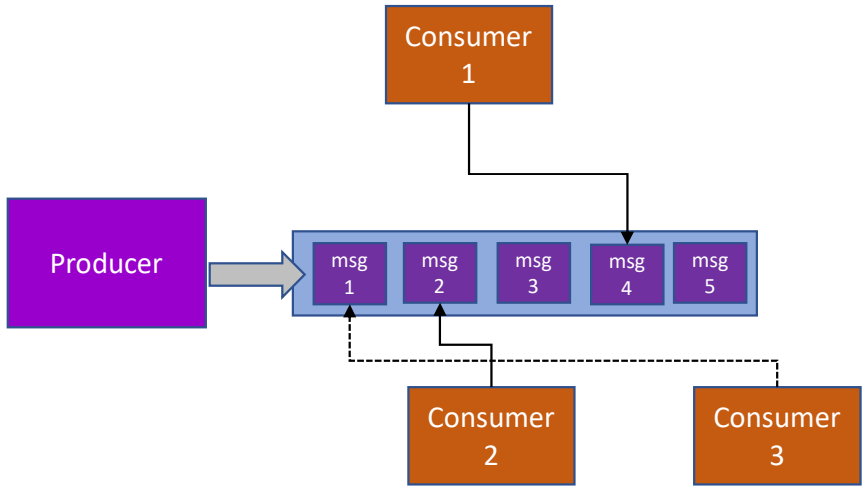




https://nielsberglund.com

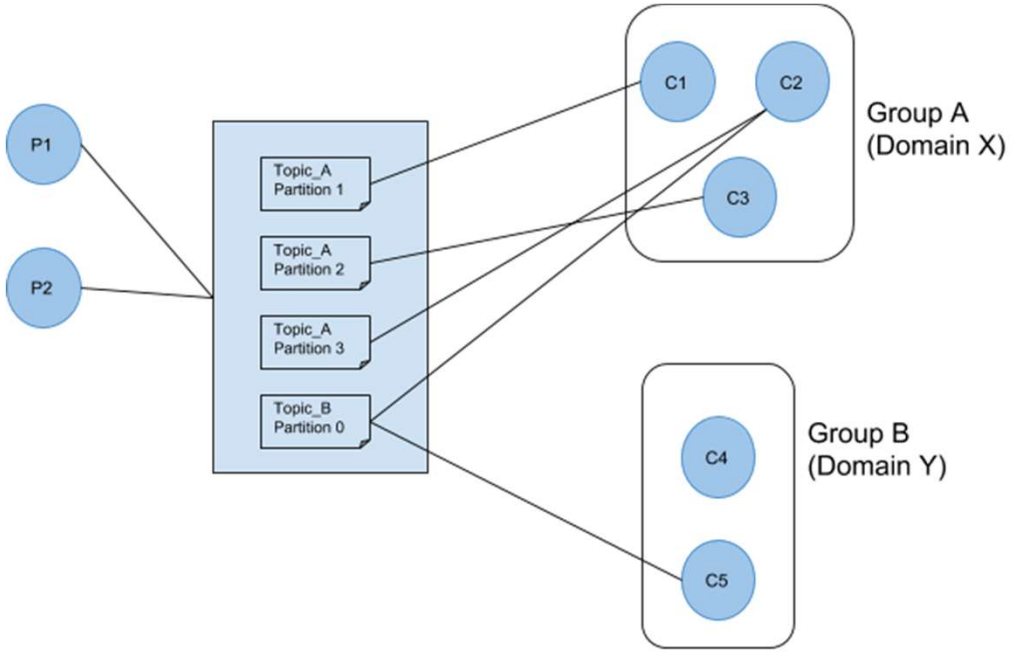
Kafka 101 - Offsets

- Offset
  - Every record within a partition assigned a sequential ID.
  - Offset unique within the partition.
  - Consumer use offsets to indicate where to start/continue read from.
  - Provides a way to replay data.



Kafka 101 - Consumer Groups

- Consumer groups
  - Group of related consumers, reading from one or more partitions in one or more topics.
  - Only one reader from one consumer group allowed to read from a partition.
  - Multiple consumer groups can read from the same partition.



Kafka & .NET

- Kafka is Scala/Java based.
  - native Java client.
- Other languages uses librdkafka.dll.
  - librdkafka high performance C implementation of the Apache Kafka client.
- Latest version of .NET client on par with Java.
- Install the client from NuGet.

```
$ mkdir test
$ cd test
$ dotnet new console -f NET5.0
# in VS Code terminal
$ dotnet add package Confluent.Kafka --version 1.7.0
```

Kafka 101 - Produce

```
using System;
using System.Threading.Tasks;
using Confluent.Kafka;

class Program
{
    public static async Task Main(string[] args)
    {
        var config = new ProducerConfig { BootstrapServers = "localhost:9092" };
        using (var p = new ProducerBuilder<string, string>(config).Build())
        {
            try
            {
                var dr = await p.ProduceAsync("test-topic", new Message<Null, string> { Value="test" });
                Console.WriteLine($"Delivered '{dr.Value}' to '{dr.TopicPartitionOffset}'");
            }
            catch (ProduceException<Null, string> e)
            {
                Console.WriteLine($"Delivery failed: {e.Error.Reason}");
            }
        }
    }
}
```

https://nielsberglund.com

Kafka 101 - Consume

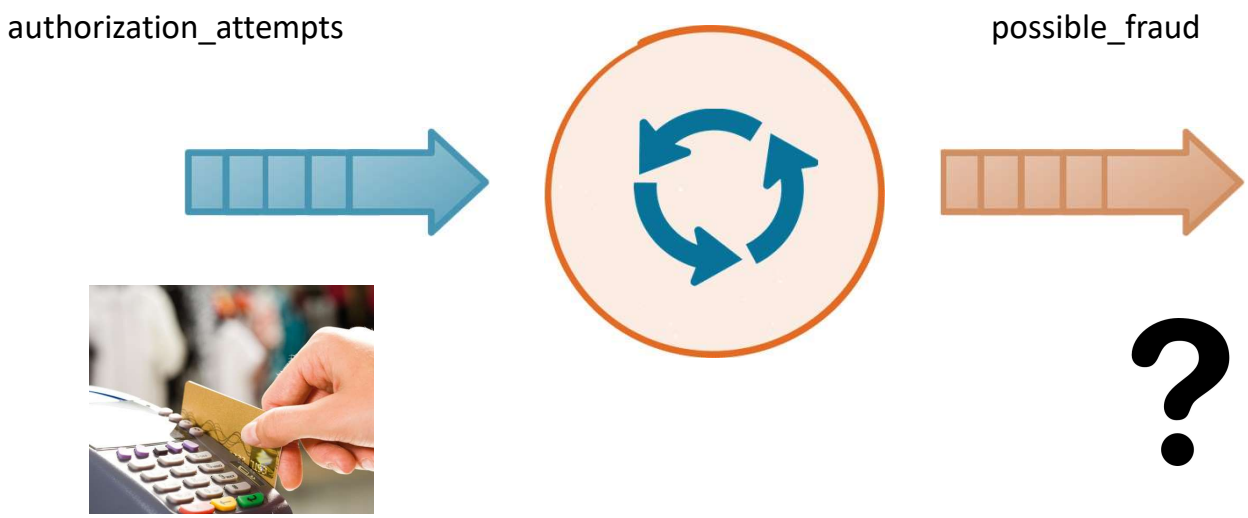
```
using System;
using System.Threading;
using Confluent.Kafka;

class Program
{
    public static void Main(string[] args)
    {
        var conf = new ConsumerConfig
        {
            GroupId = "test-consumer-group",
            BootstrapServers = "localhost:9092",
            AutoOffsetReset = AutoOffsetReset.Earliest
        };

        using (var c = new ConsumerBuilder<Ignore, string>(conf).Build())
        {
            c.Subscribe("my-topic");

            try
            {
                while (true)
                {
                    try
                    {
                        //cancellationtoken cts created above somewhere
                        var cr = c.Consume(cts.Token);
                        Console.WriteLine($"Consumed message '{cr.Message.Value}' at: '{cr.TopicPartitionOffset}'.");
                    }
                    catch (ConsumeException e)
                    {
                        Console.WriteLine($"Error occured: {e.Error.Reason}");
                    }
                }
            }
            catch (OperationCanceledException)
            {
                c.Close();
            }
        }
    }
}
```

Stream Processing

- Stream processing is what we do when we want to handle events

https://nielsberglund.com

Stream Processing - I



- Possible fraud = Authorization attempts > 3 within x timeframe

How to Implement Stream Processing

- Roll your own using Kafka consumer ☹
  - how to handle state
- KStreams introduced 2016.
  - library for building streaming applications, specifically applications that transform input Kafka topics into output Kafka topics.
  - Scala/Java only ☹
- KSQL introduced 2017
  - SQL Syntax. No Scala/Java! ☺

ksqlDB

- Evolution of KSQL, introduced 2019.
- Event streaming db.
- Allows you to build stream processing applications.
- Write real-time applications in SQL (no Java, Scala, etc.).
  - For complex functionality write User Defined Functions when needed.
- Lowers the bar for implementing streaming.

Lower the Bar for Streaming - I

ksqlDB

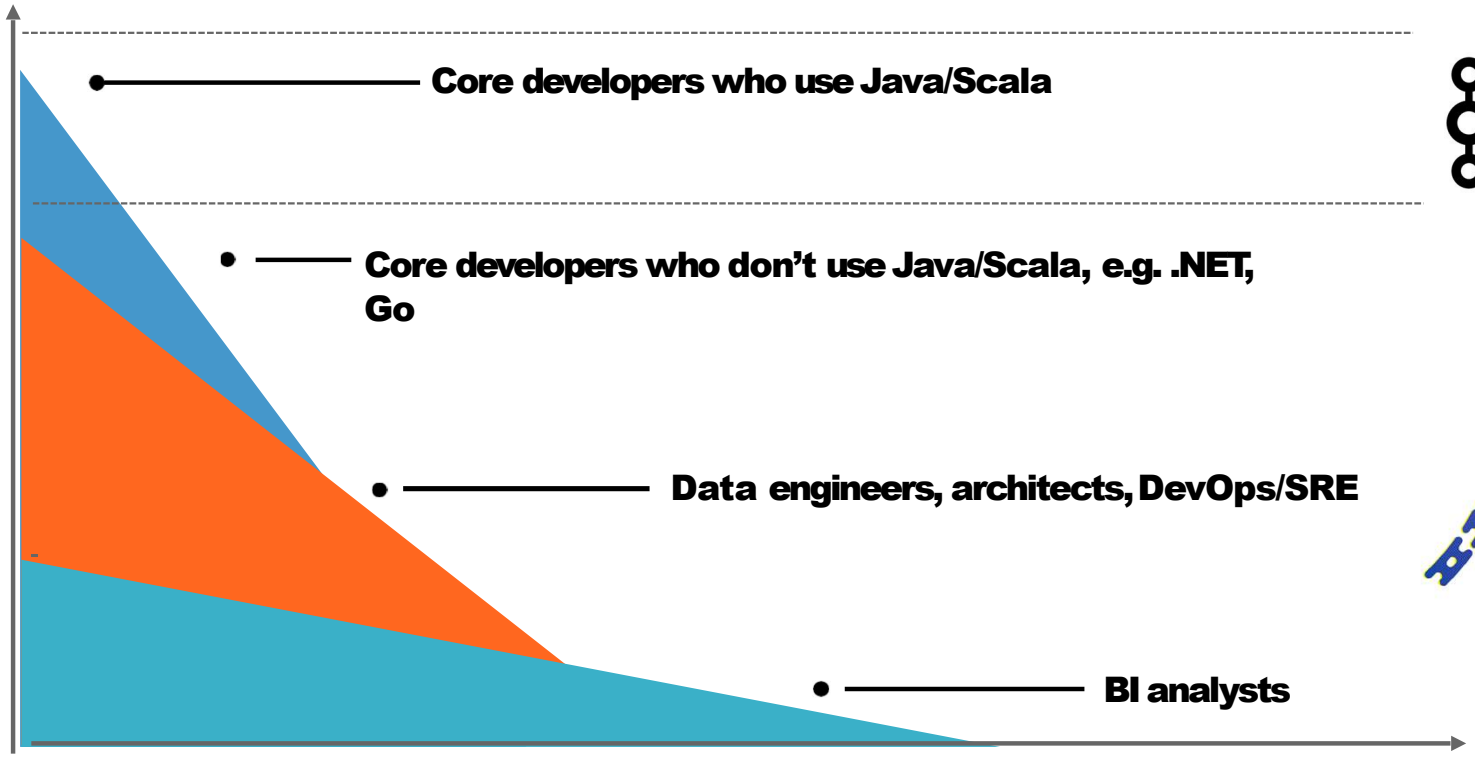
```
CREATE STREAM
fraudulent_payments AS
SELECT * FROM payments
WHERE fraudProbability > 0.8;
```

kafka streams

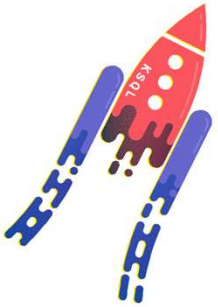
```
object FraudFilteringApplication extends App {

  val config = new java.util.Properties
  config.put(StreamsConfig.APPLICATION_ID_CONFIG, "fraud-filtering
  config.put(StreamsConfig.BOOTSTRAP_SERVERS_CONFIG, "kafka-broker

  val builder: StreamsBuilder = new StreamsBuilder()
  val fraudulentPayments: KStream[String, Payment] = builder
    .stream[String, Payment]("payments-kafka-topic")
    .filter((_ ,payment) => payment.fraudProbability > 0.8)

  val streams: KafkaStreams = new KafkaStreams(builder.build(), co
  streams.start()
}
```

Source: https://www.slideshare.net/ConfluentInc/introduction-to-ksql-streaming-sql-for-apache-kafka

Lower the Bar for Streaming - II



kafka streams



https://nielsberglund.com

ksqlDB - Streams and Tables

- Stream - unbounded sequence of structured data (events).
  - Immutable.
- Table - view of a stream or another table.
  - Mutable.
- Stream & table duality
  - Turn a stream into a table via DML manipulations: `SELECT`, `COUNT()`, `SUM()`, etc.
  - Turn a table into a stream by capturing inserts, updates, and deletes.

ksqlDB - Syntax

- Semantics similar to ANSI SQL.
- Statements:
  - `CREATE` … (stream, table, and more).
    - Is combined with `SELECT/INSERT.`
    - Supports windowing functions, `(SESSION, HOPPING, TUMBLING).`
  - `INSERT` …
  - `SELECT` (more later).
  - and more.
- Functions:
  - Scalar.
  - Aggregation.
  - Table.

ksqlDB - Push & Pull Queries

- Pull:s the current value and terminates.

```
SELECT *
FROM wagers
WHERE ROWKEY = '12345';
```

- Push:es a continuous stream of updates to a ksqlDB stream or table.

```
SELECT *
FROM wagers
EMIT CHANGES;
```

ksqlDB - Examples

```
CREATE STREAM fraudulent_payments
AS
SELECT * FROM payments
WHERE fraudProbability > 0.8;
```

```
CREATE STREAM syslog_invalid_users
AS
SELECT host, message
FROM syslog
WHERE message LIKE '%Invalid user%';
```

```
CREATE TABLE possible_fraud
AS
SELECT card_number, COUNT(*)
FROM authorization_attempts
WINDOW TUMBLING (SIZE 5 SECONDS)
GROUP BY card_number
HAVING COUNT(*) > 3;
```

Summary

- Kafka distributed streaming platform.
- Topics, partitions, offsets, and consumer groups.
- Fully fledged .NET Client
- Stream processing handling the messages that arrives in Kafka
- ksqlDB lowers the barrier for entry to write real-time stream processing applications.

Thanks!

Questions?

Niels Berglund
niels.it.berglund@gmail.com
https://nielsberglund.com
https://twitter.com/nielsberglund