NOVATEC

Agile Integration

PA Agile Integration, Kafka and Event-Streaming

Oliver Berger, Nadja Hagen

Tobias Dehn, Christopher Weiß, Uwe Eisele

Content

- Integration
- Event-Streaming Platform Kafka
- Asynchronous Communication
- Kafka Basics & Components
- Kafka APIs
- Kafka Examples
- Kafka Exercises

NOVATEC

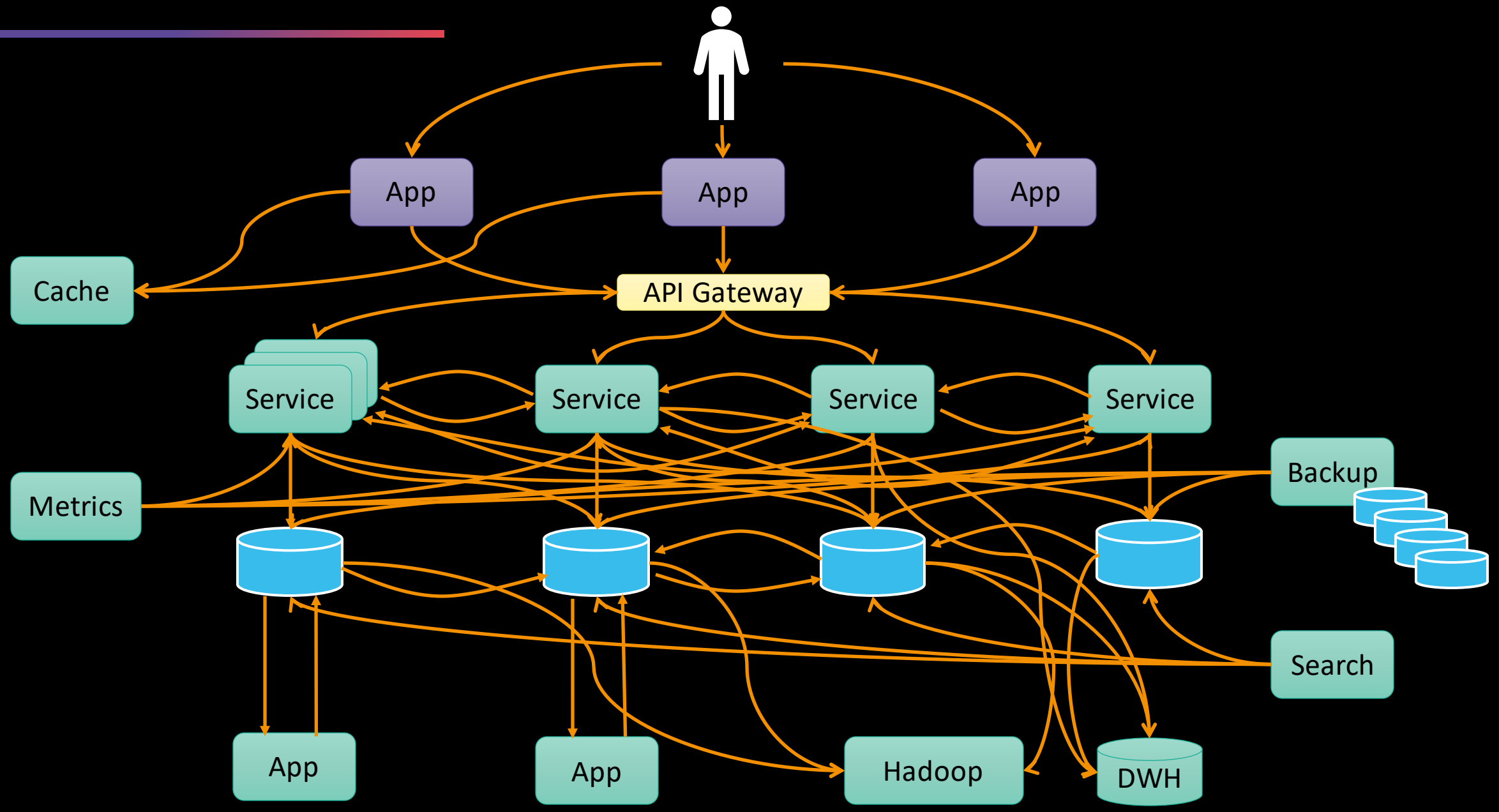
Integration

..of distributed systems is a complex challenge

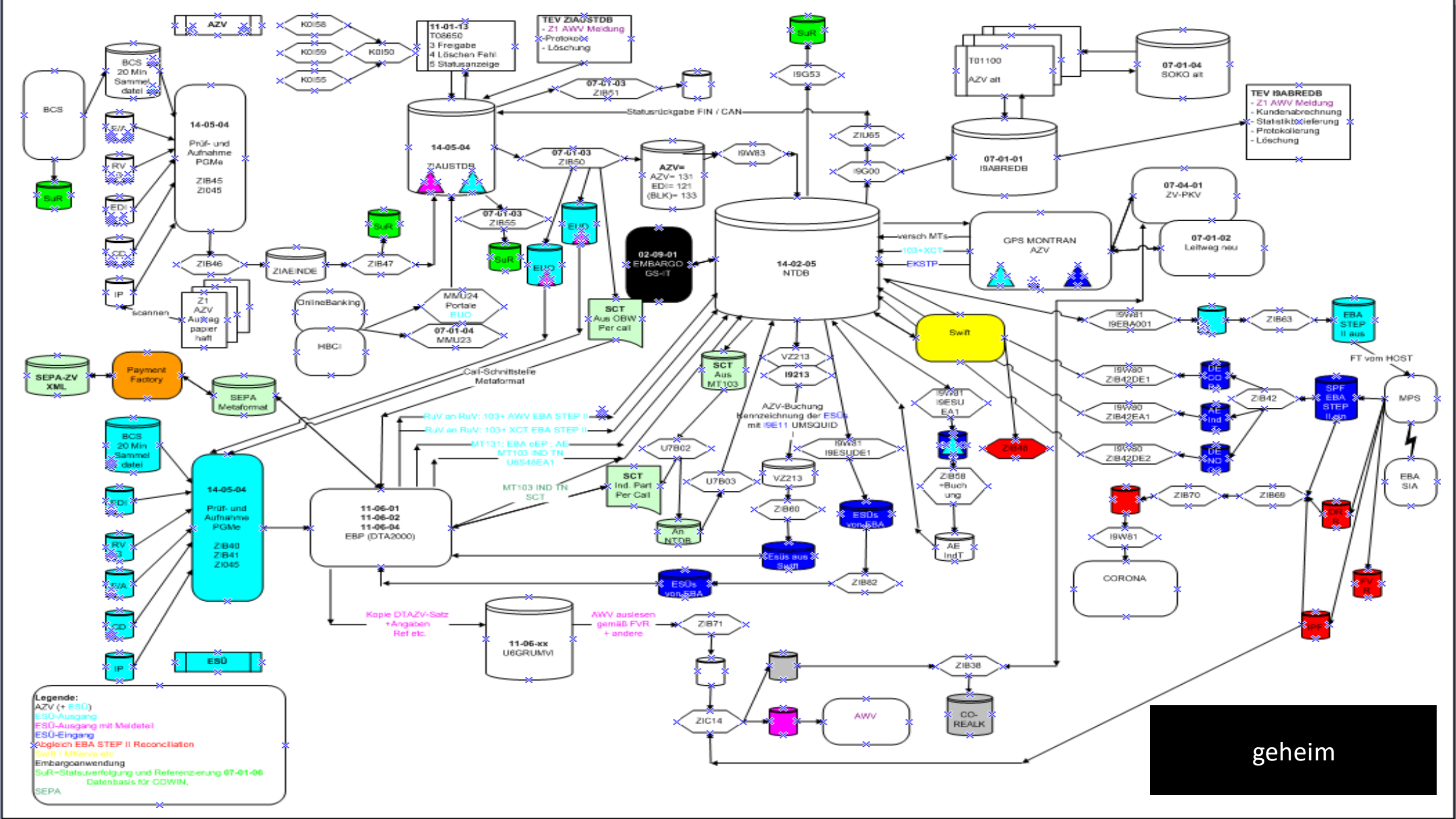
NOVATEC

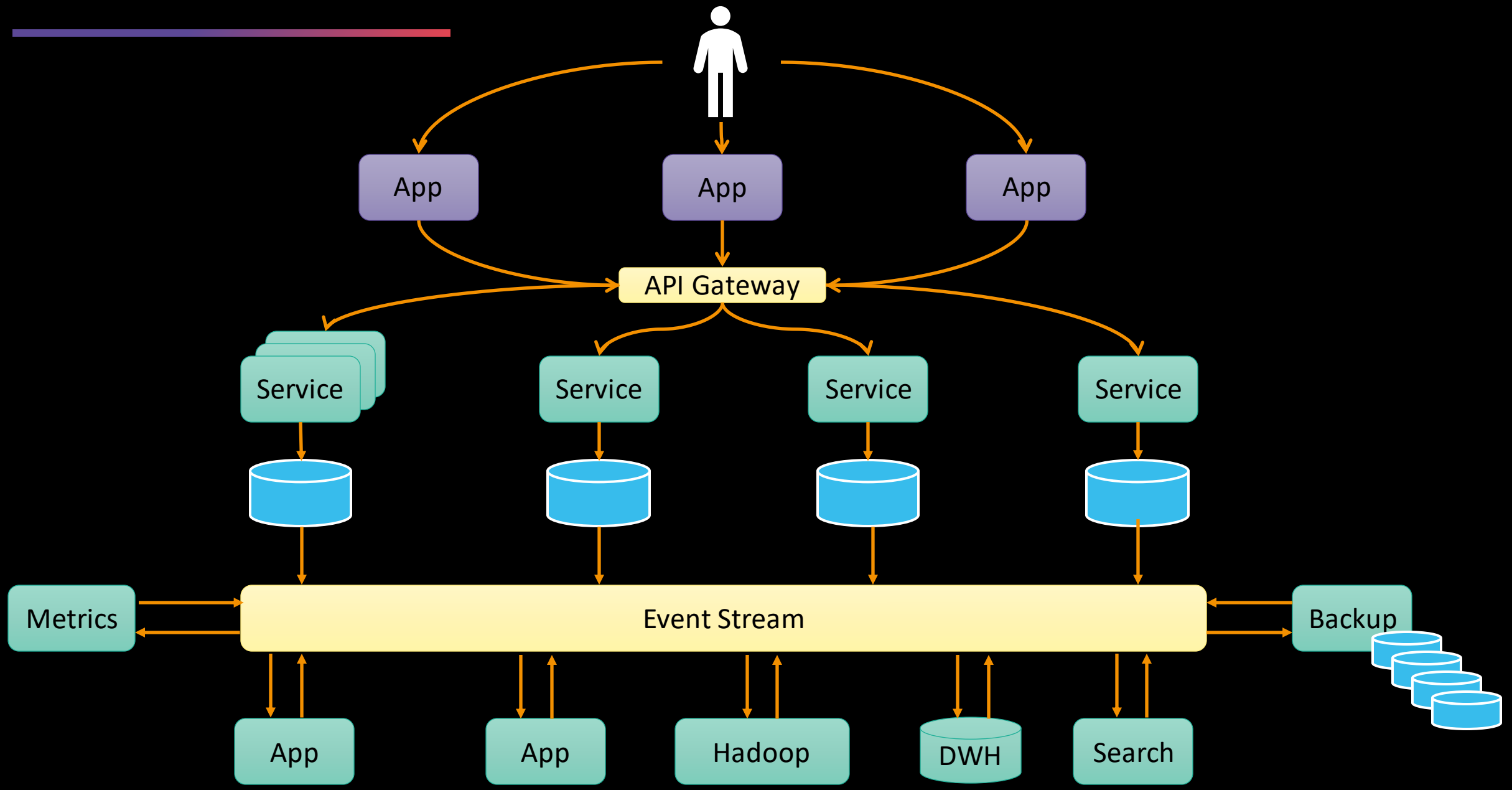
Integration in Distributed Systems, complex challenge lots of things to consider

Technologies	Programming Language	Application Architecture	Communication Paradigms
• Standards (SOAP, REST , JMS) • Data formats (JSON, XML, Avro) • Frameworks • Proprietary Interfaces	• Java • C, C#, .Net • Python • Cobol	• Client Server • Monolith • SOA • Microservices • Serverless	• Batch • Realtime • Request-Response • Pub-Sub • Fire & Forget


App
App
App
Cache
API Gateway
Service
Service
Service
Service
Backup
Metrics
Search
App
App
Hadoop
DWH

NOVATEC


AZV
K0I58
K0I59
K0I55
K0I50
11-01-13
T08650
3 Freigabe
4 Löschen Fehl
5 Statusanzeige
TEV ZIAUSTDB
- Z1 AWV Meldung
- Protokoll
- Löschung
SuR
I9G53
T01100
AZV alt
07-01-04
SOKO alt
TEV I9ABREDB
- Z1 AWV Meldung
- Kundenabrechnung
- Statistiklieferung
- Protokollierung
- Löschung
BCS
BCS
20 Min
Sammel
datei
14-05-04
Prüf- und
Aufnahme
PGMe
ZIB45
ZI045
07-01-03
ZIB51
Statusrückgabe FIN / CAN
14-05-04
ZIAUSTDB
07-01-03
ZIB50
AZV=
AZV= 131
EDI= 121
(BLK)= 133
I9W83
ZIU65
I9G00
07-01-01
I9ABREDB
07-04-01
ZV-PKV
07-01-02
Leitweg neu
07-01-03
ZIB55
EUO
02-09-01
EMBARGO
GS-IT
14-02-05
NTDB
versch MTs
103+XCT
EKSTP
GPS MONTRAN
AZV
ZIB46
ZIAEINDE
ZIB47
Z1
AZV
Auxag
papier
haft
scannen
OnlineBanking
HBCI
MMU24
Portale
EUO
07-01-04
MMU23
SCT
Aus OBW
Per call
Swift
I9W81
I9EBA001
ZIB63
EBA
STEP
II aus
FT vom HOST
SEPA-ZV
XML
Payment
Factory
SEPA
Metaformat
Call-Schnittstelle
Metaformat
SCT
Aus
MT103
VZ213
I9213
AZV-Buchung
Kennzeichnung der ESÜs
mit I9E11 UMSQUID
I9W81
I9ESU
EA1
I9W90
ZIB42DE1
I9W90
ZIB42EA1
I9W90
ZIB42DE2
ZIB42
SPF
EBA
STEP
II an
MPS
EBA
SIA
RuV an RuV: 103+ AWV EBA STEP II
RuV an RuV: 103+ XCT EBA STEP II
MT131: EBA cEP , AE
MT103 IND TN
U6S48EA1
U7B02
U7B03
I9W81
I9ESUDE1
ZIB48
ZIB58
+Buch
ung
AE
IndT
MT103 IND TN
SCT
SCT
Ind. Part
Per Call
VZ213
ZIB60
An
NTDB
ESÜs
von EBA
ESÜs aus
Swift
ZIB62
ZIB70
ZIB69
I9W81
CORONA
BCS
20 Min
Sammel
datei
14-05-04
Prüf- und
Aufnahme
PGMe
ZIB40
ZIB41
ZI045
11-06-01
11-06-02
11-06-04
EBP (DTA2000)
ESÜ
Kopie DTAZV-Satz
+Angaben
Ref etc.
11-06-xx
U6GRUMVI
AWV auslesen
gemäß FVR
+ andere
ZIB71
ZIB38
ZIC14
AWV
CO-
REALK
IPF
Legende:
AZV (+ ESÜ)
ESÜ-Ausgang
ESÜ-Ausgang mit Meldetel
ESÜ-Eingang
Abgleich EBA STEP II Reconciliation
Swift / MKerva etc
Embargoanwendung
SuR=Statusverfolgung und Referenzierung 07-01-06
Datenbasis für COWIN,
SEPA
geheim


App
App
App
API Gateway
Service
Service
Service
Service
Metrics
Event Stream
Backup
App
App
Hadoop
DWH
Search

NOVATEC

Content

- Integration
- Event-Streaming Platform Kafka
- Asynchronous Communication
- Kafka Basics & Components
- Kafka APIs
- Kafka Examples
- Kafka Exercises

NOVATEC

Event-Streaming Platform Kafka

Why should you be interested?

NOVATEC

Apache Kafka: The Event Streaming Platform

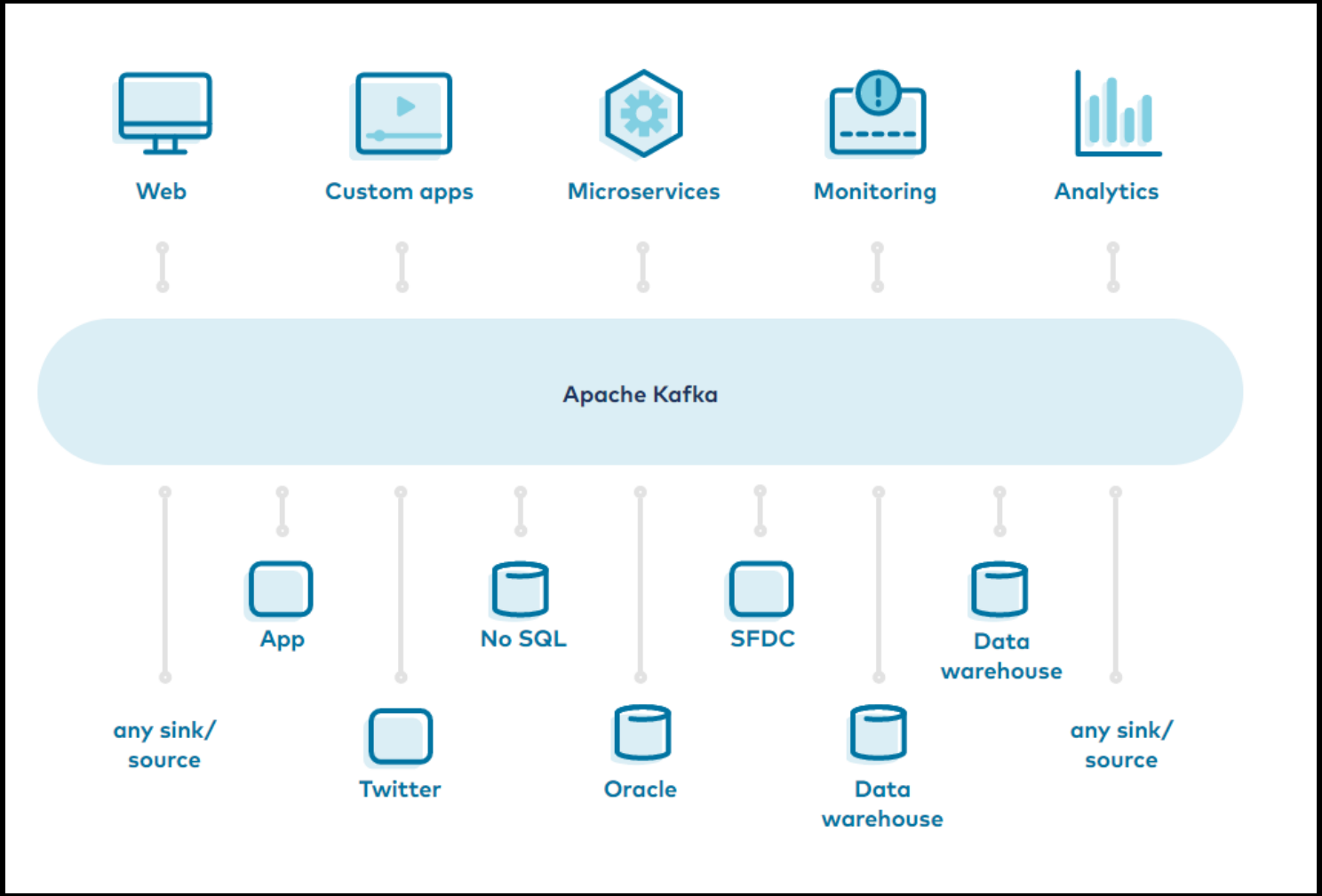
- developed at LinkedIn in 2011 and made open source
- can process several trillion (10^12) events per day
- originally designed as a messaging queue
- based on an abstraction of a distributed commit log
- evolved from a messaging queue to a full-fledged event streaming platform
- de facto standard for Event-Streaming Platform (> 95% of Event-Streaming-Projects rely on kafka)

Thousends of Enterprises rely on Kafka and Event-Streaming

Source: https://kafka.apache.org/powered-by

NOVATEC

Possible Applications of Apache Kafka



NOVATEC

3 Key Functionalities of a Streaming Platform

Pub-Sub

Store

Process

Content

- Integration
- Event-Streaming Platform Kafka
- Asynchronous Communication
- Kafka Basics & Components
- Kafka APIs
- Kafka Examples
- Kafka Exercises

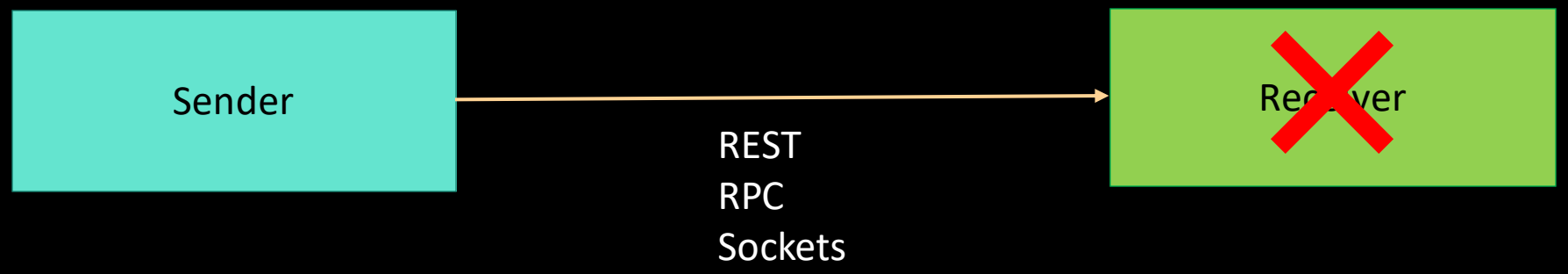
NOVATEC

Asynchronous Communication

Why do you need a Messaging System?

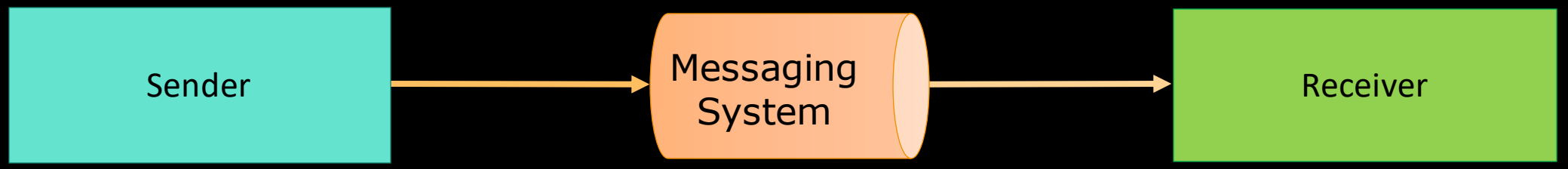
NOVATEC

Why do you need a Messaging System ?



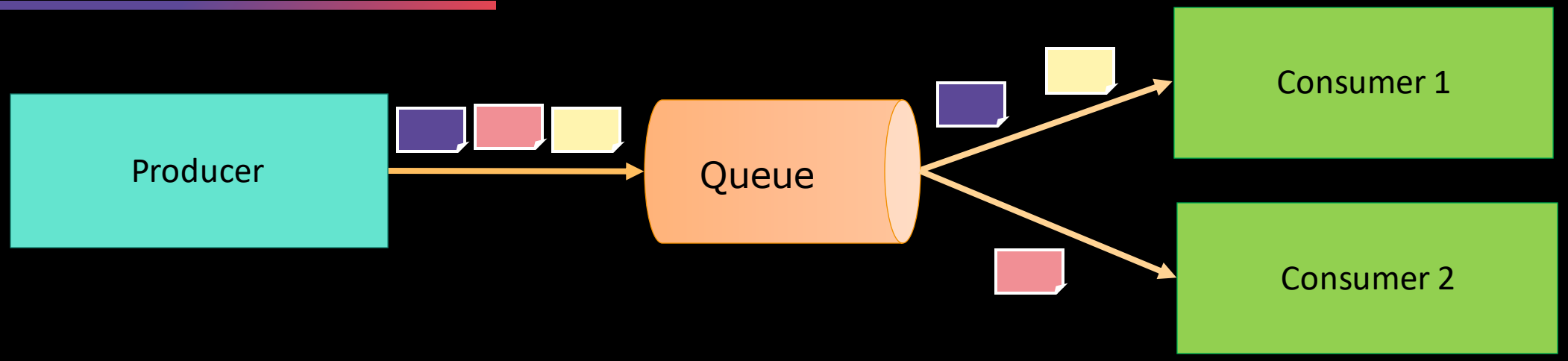
- Challenge 1: Availability
- Challenge 2: Processing Velocity
- Challenge 3: Processing Acknowledgement

Solution: Messaging



- Decoupling via Messaging
- Examples for Messaging-Systems:
  - MQ-Series,
  - JMS-Messaging (ActiveMQ, Rabbit-MQ),
  - Kafka.
- Transfermode: Queue or Topic

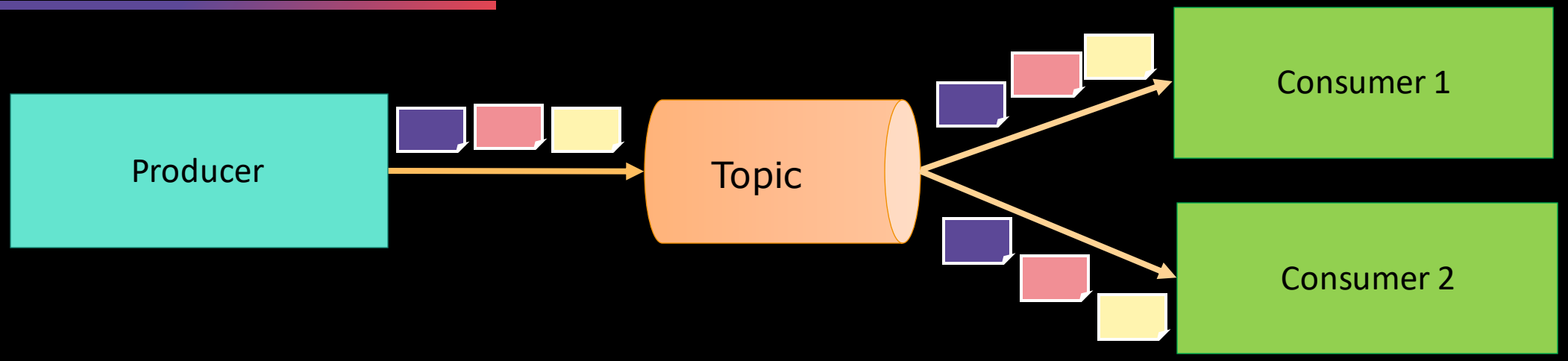
Queue Mode
1 to 1 Topologie



- each message gets processed exactly once

NOVATEC

Topic Mode
1 to n Topologie



- each message can be consumed by independeant consumers
- each consumer receives all messages after subscription
- sequence of messages is guaranteed
- only new messages are delivered

NOVATEC

Content

- Integration
- Event-Streaming Platform Kafka
- Asynchronous Communication
- Kafka Basics & Components
- Kafka APIs
- Kafka Examples
- Kafka Exercises

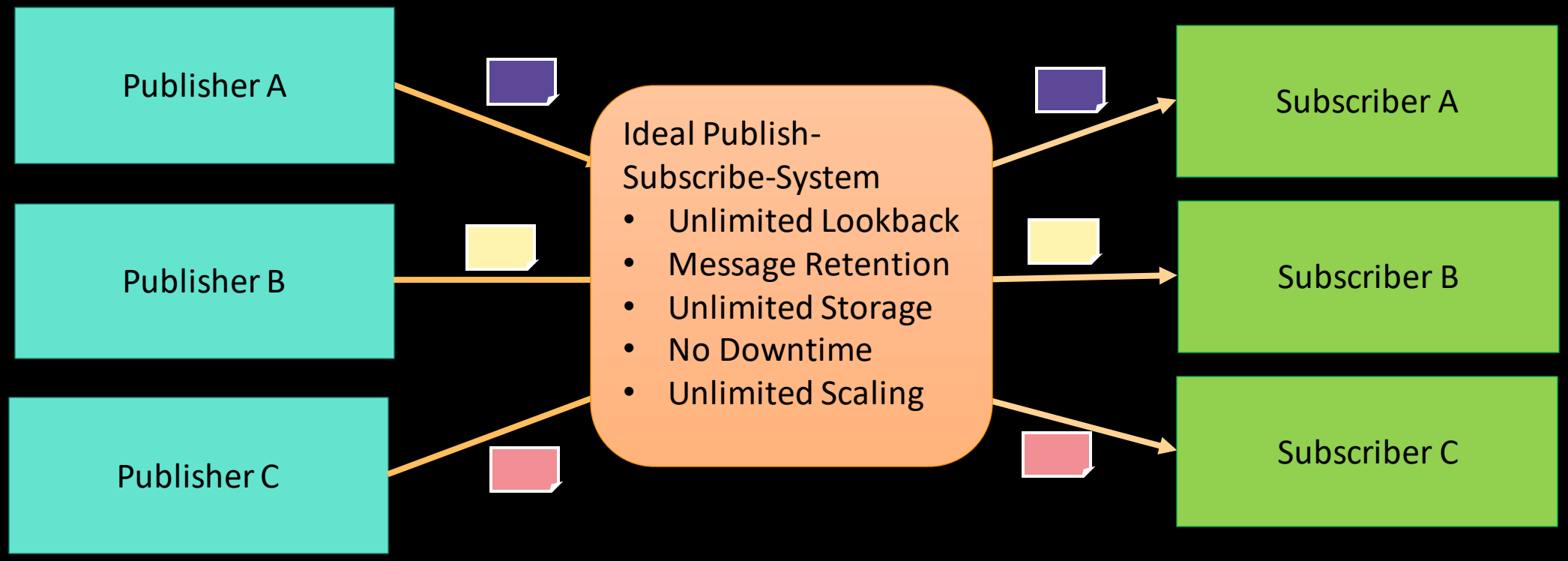
NOVATEC

Kafka

Basics & Components

NOVATEC

How would an ideal Publish-Subscribe System look like?

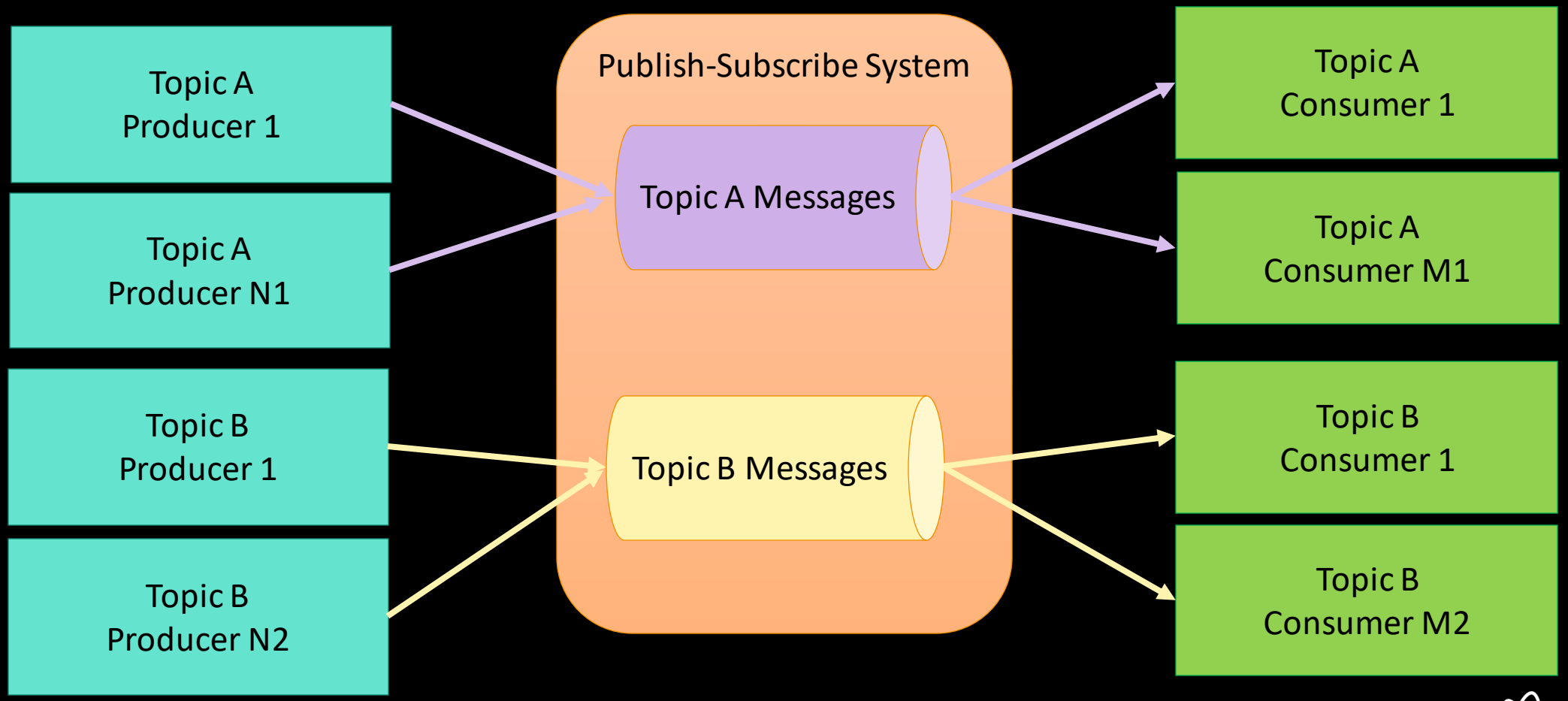


NOVATEC

Kafka Architecture in Comparison to the ideal Pub-Sub System

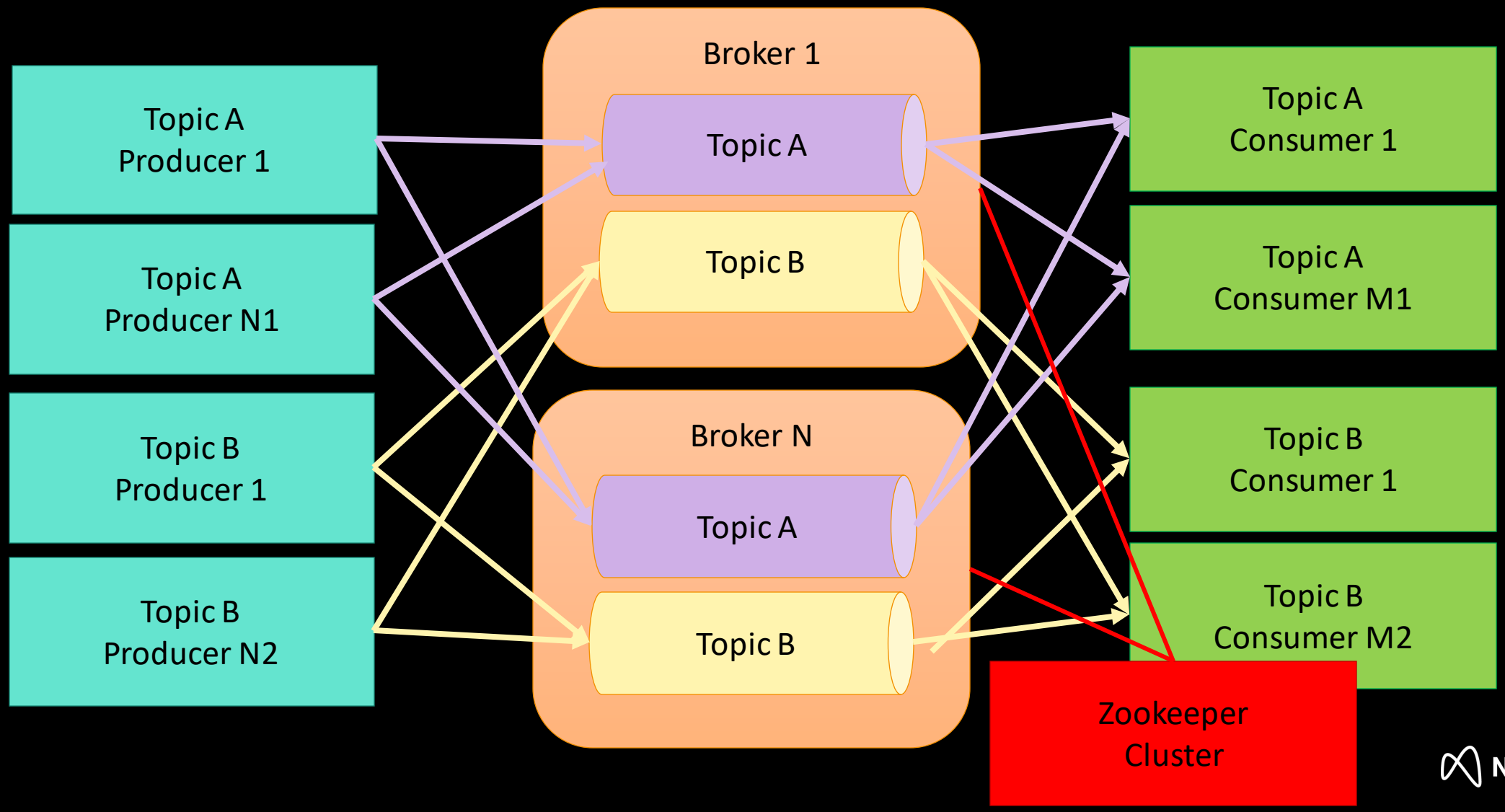
- Key Differences
  - Messaging is implemented on top of a replicated, distributed unmutable commit log.
  - The client has more functionality and, therefore, more responsibility.
  - Messaging is optimized for batches instead of individual messages.
  - Messages are retained even after they are consumed; they can be consumed again.
- Consequences of these Design Decisions
  - extreme horizontal scalability
  - very high throughput
  - high availability
  - but different semantics and message delivery guarantees

Topics in a Publish-Subscribe System



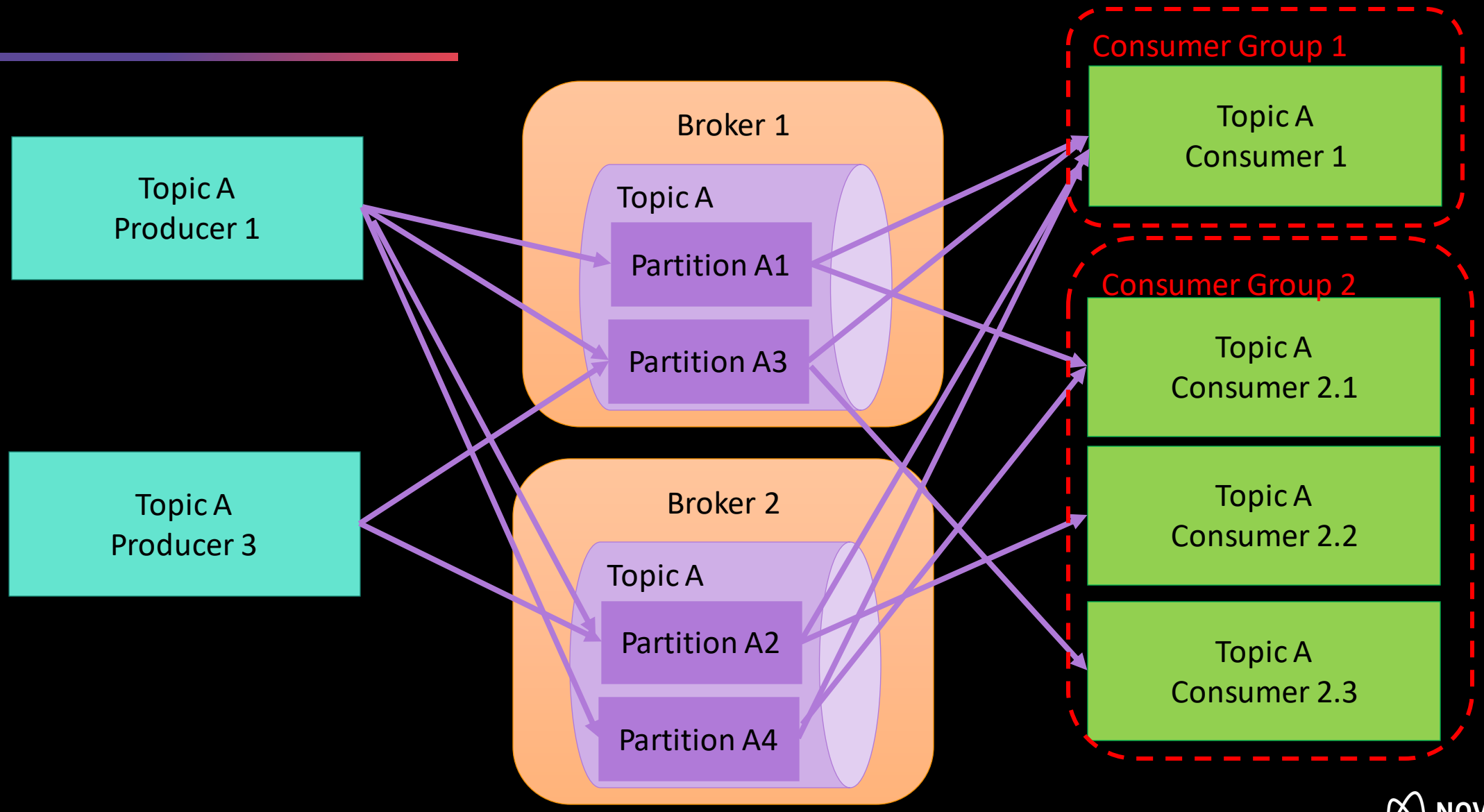
NOVATEC

Broker in a Publish-Subscribe System



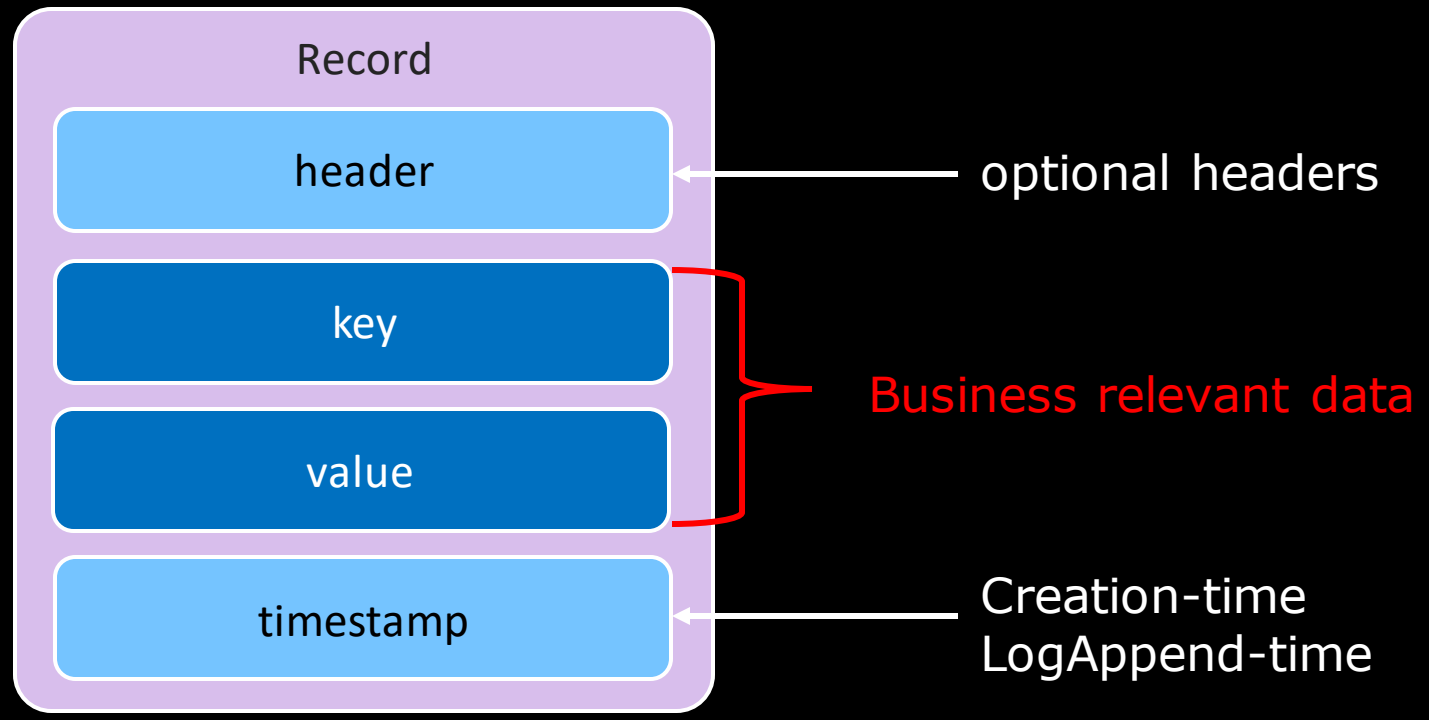
NOVATEC

Partitions: Partition Count 4



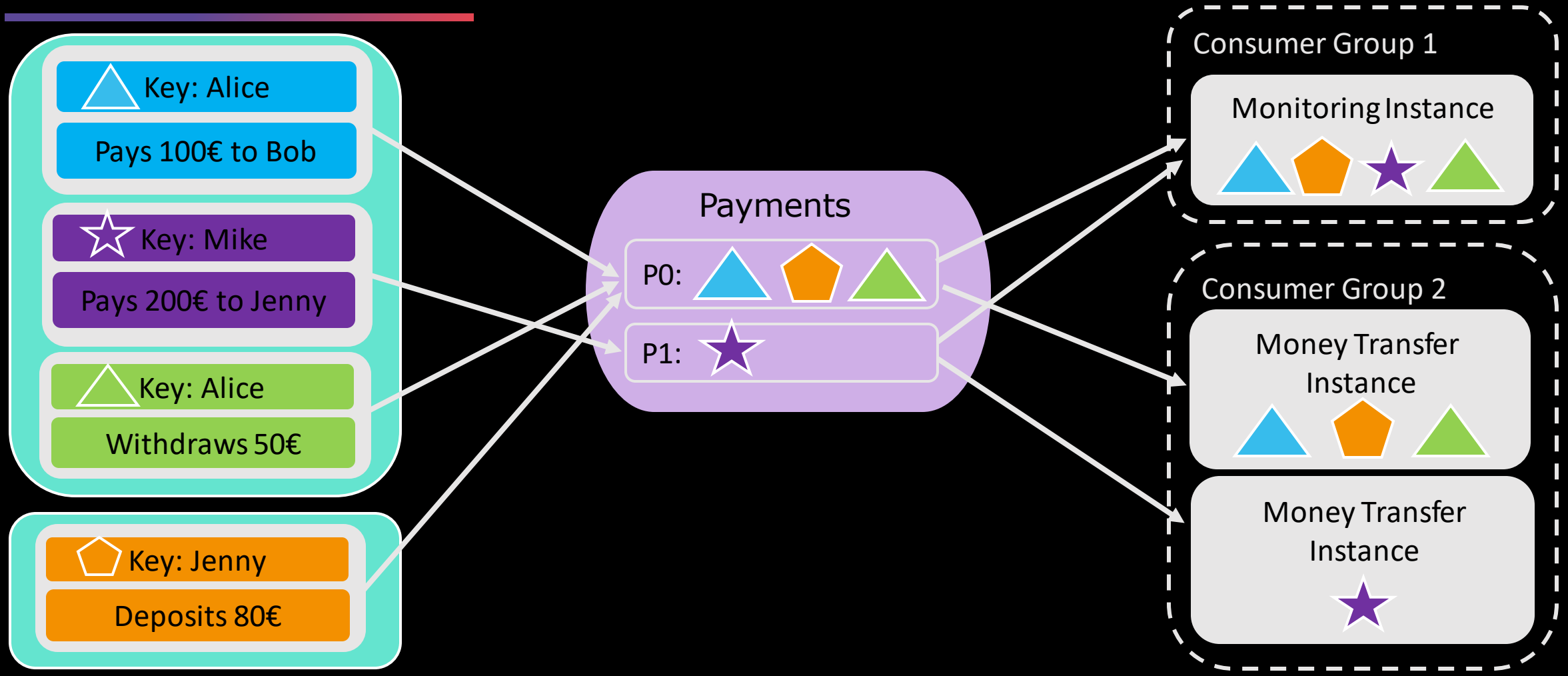
NOVATEC

The Record – The Atomic Unit of Kafka
Synonyms: Message or Event



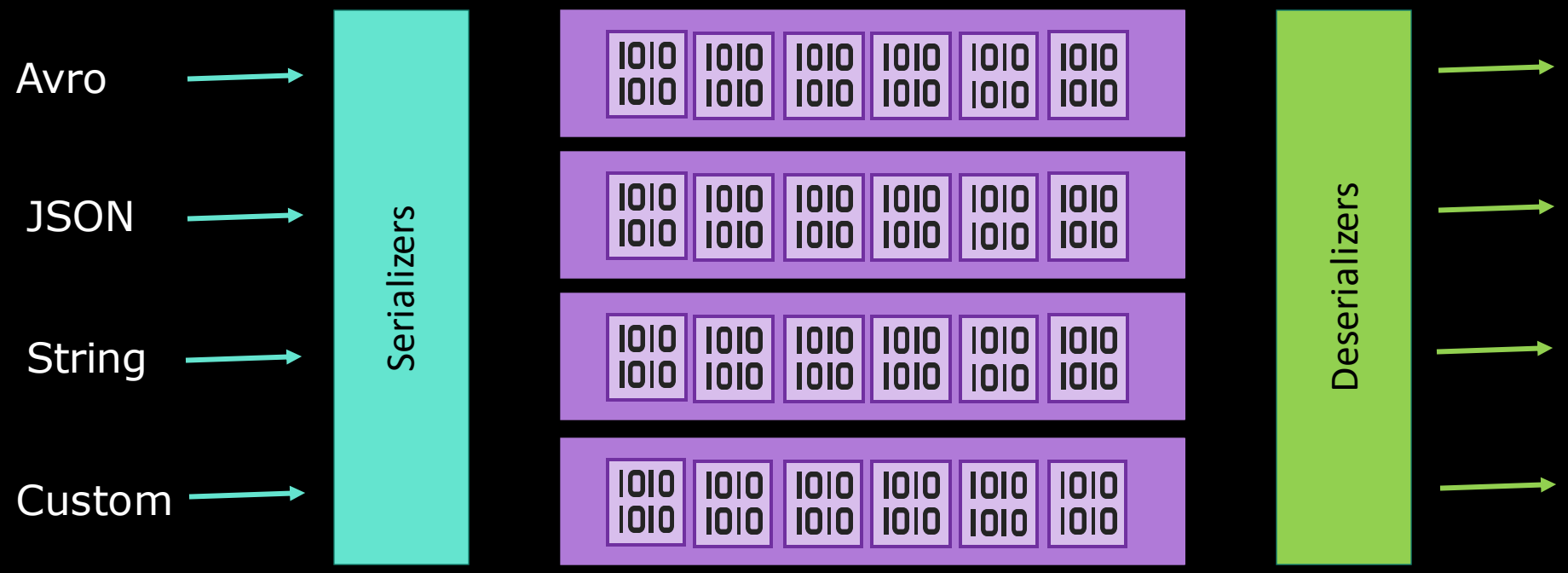
NOVATEC

Example: Payment Processing



NOVATEC

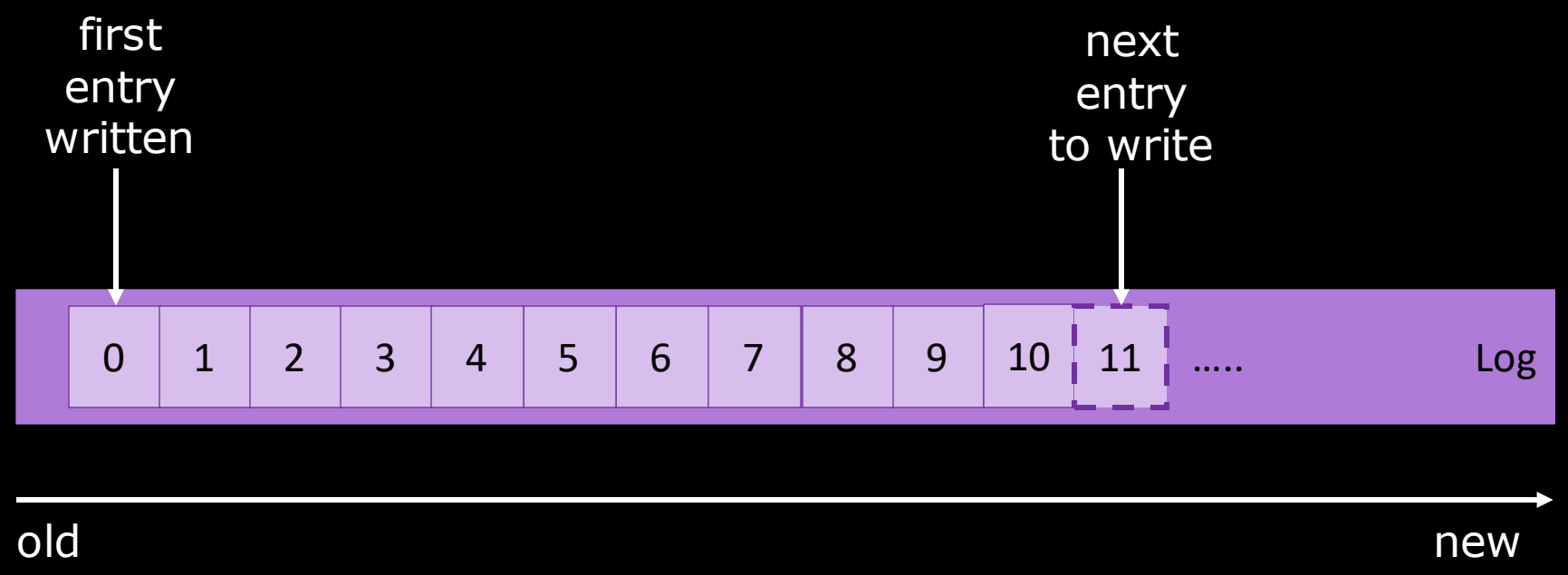
Serialisation
Kafka stores Byte Arrays



NOVATEC

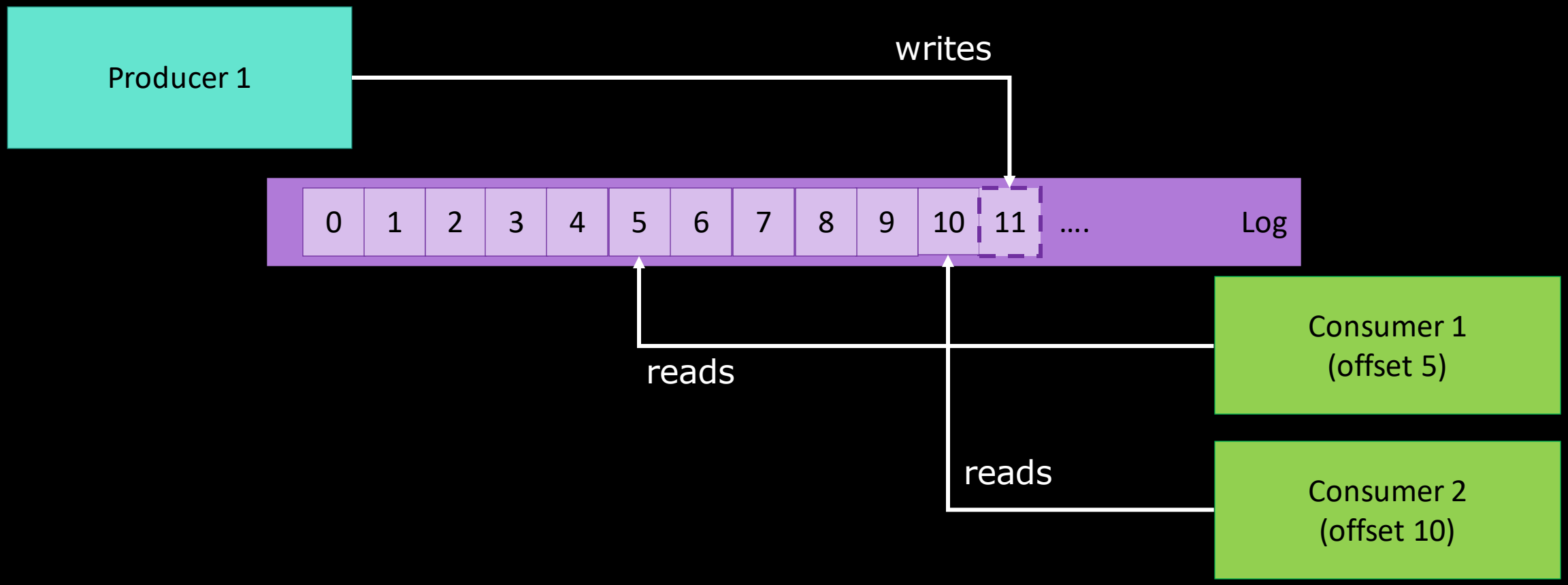
Kafka Commit Log
Abstraction to understand Streaming

- Immutable, append-only data structure (record,event)
- Offset: the position of the record/event in the log



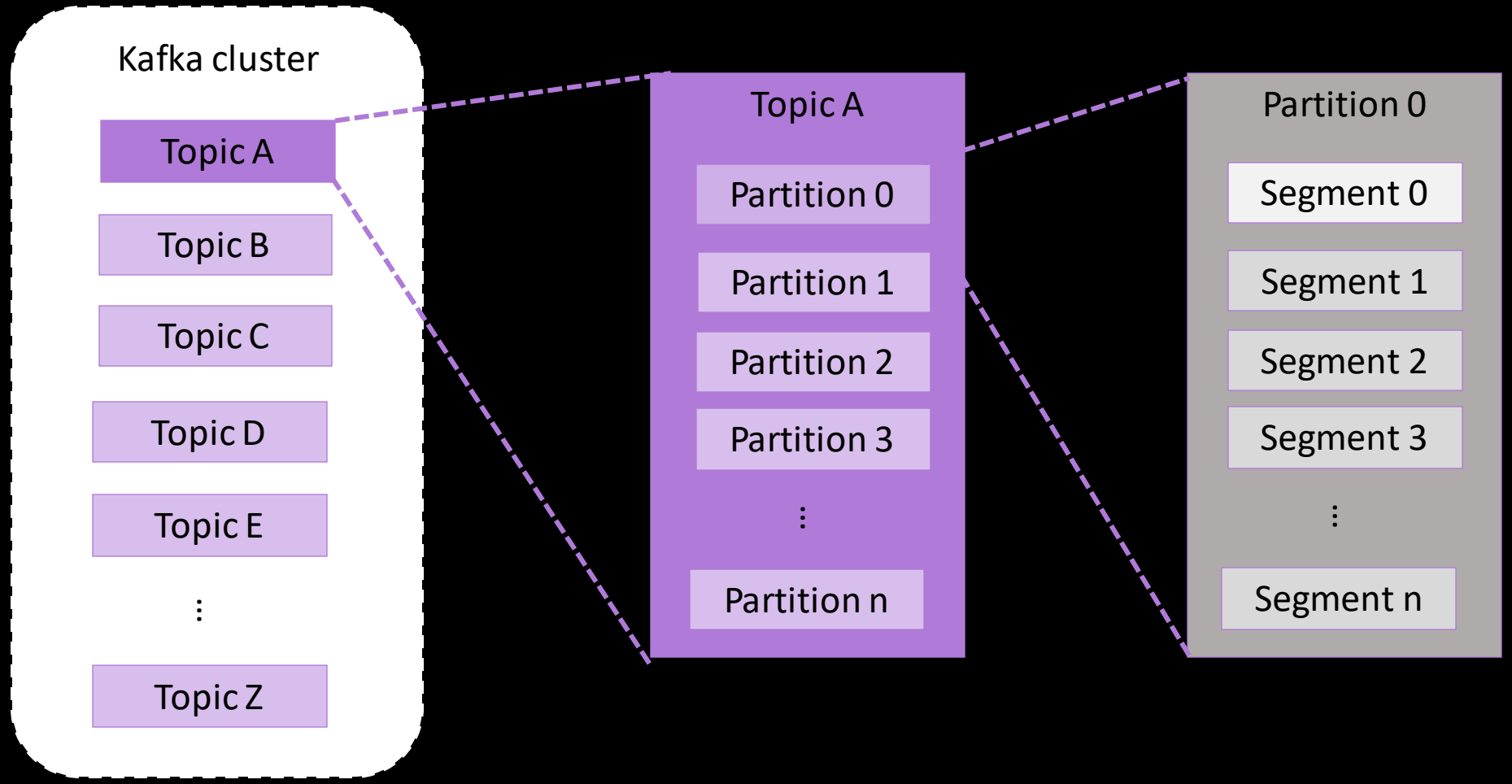
NOVATEC

Decoupling Data Producers from Data Consumers



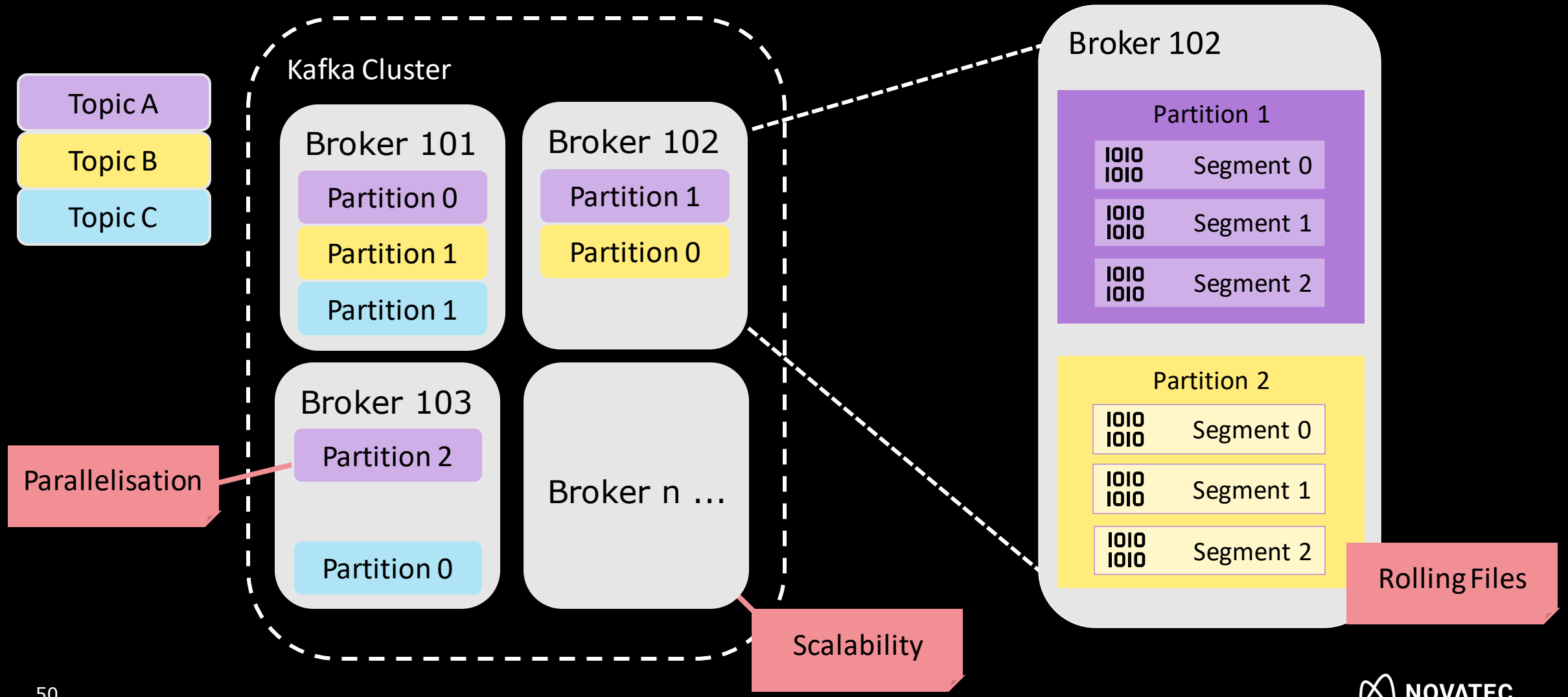
NOVATEC

Logical View of Topics, Partitions & Segments
Each Partition is a Commit Log



NOVATEC

Physical View of Topics, Partitions & Segments



NOVATEC

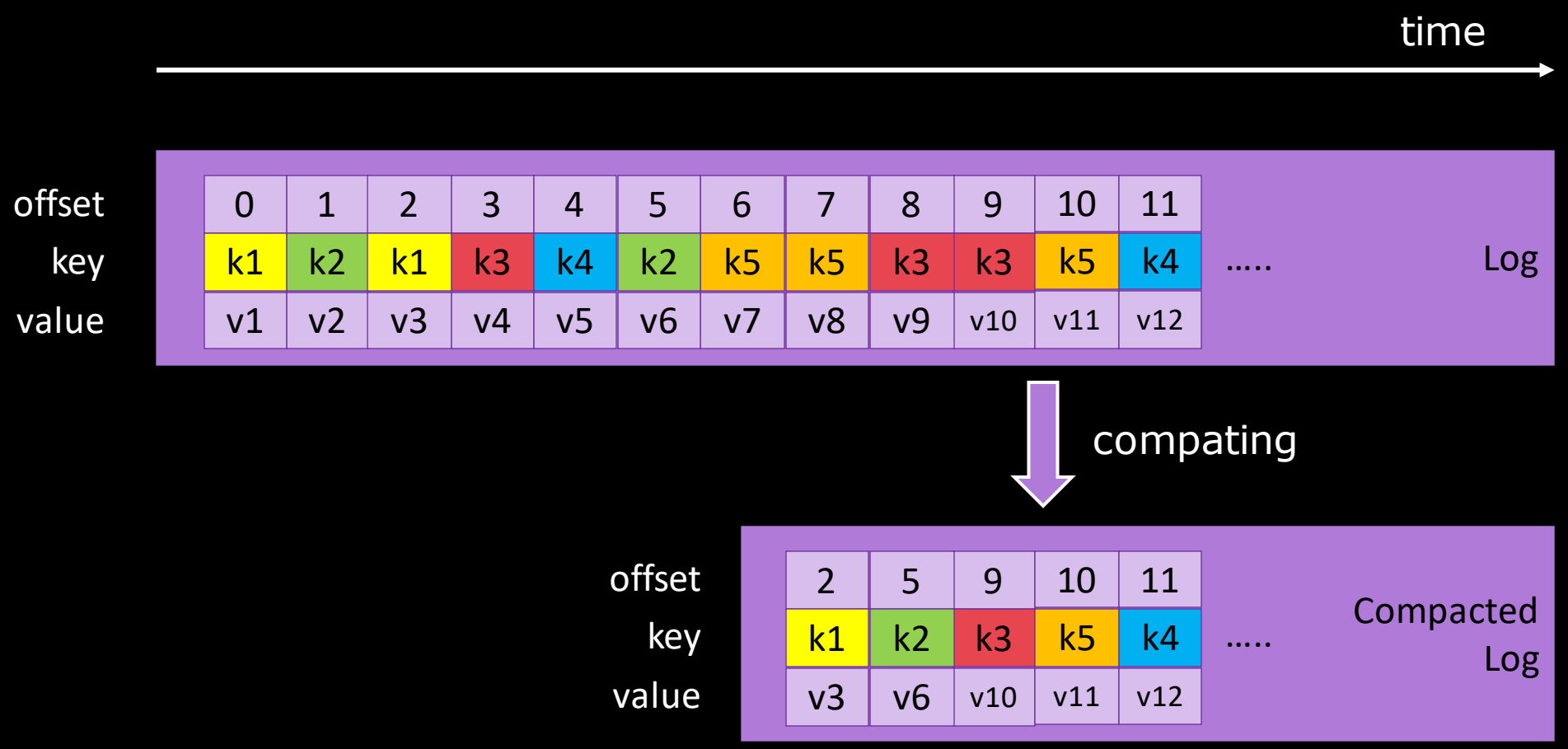
Brokers in several Data Centers

NOVATEC

Managing Log File Growth Retention-Policies

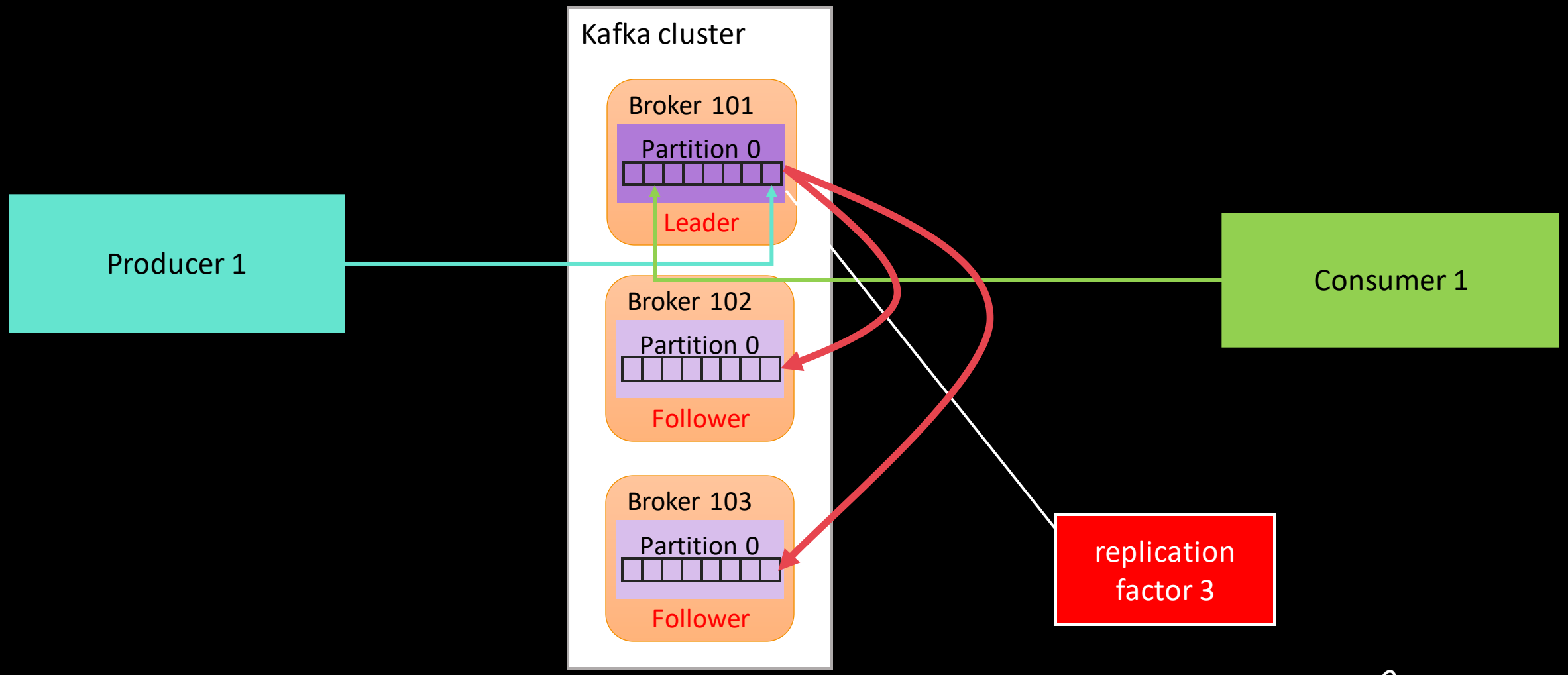
- Cleanup.policy
  - delete (default)
    - Segments too old: retention.ms (default 7 days)
    - Partitions too large: retention.bytes (default: -1 unlimited)
  - compact (keep only the freshest value
  - delete and compact (Example: Order – Management)
- Cleanup applies to Segment-Files.
  - messages are guaranteed to live at least as long as retention time
  - only non-active segment files get deleted upon Cleanup

Log Compaction



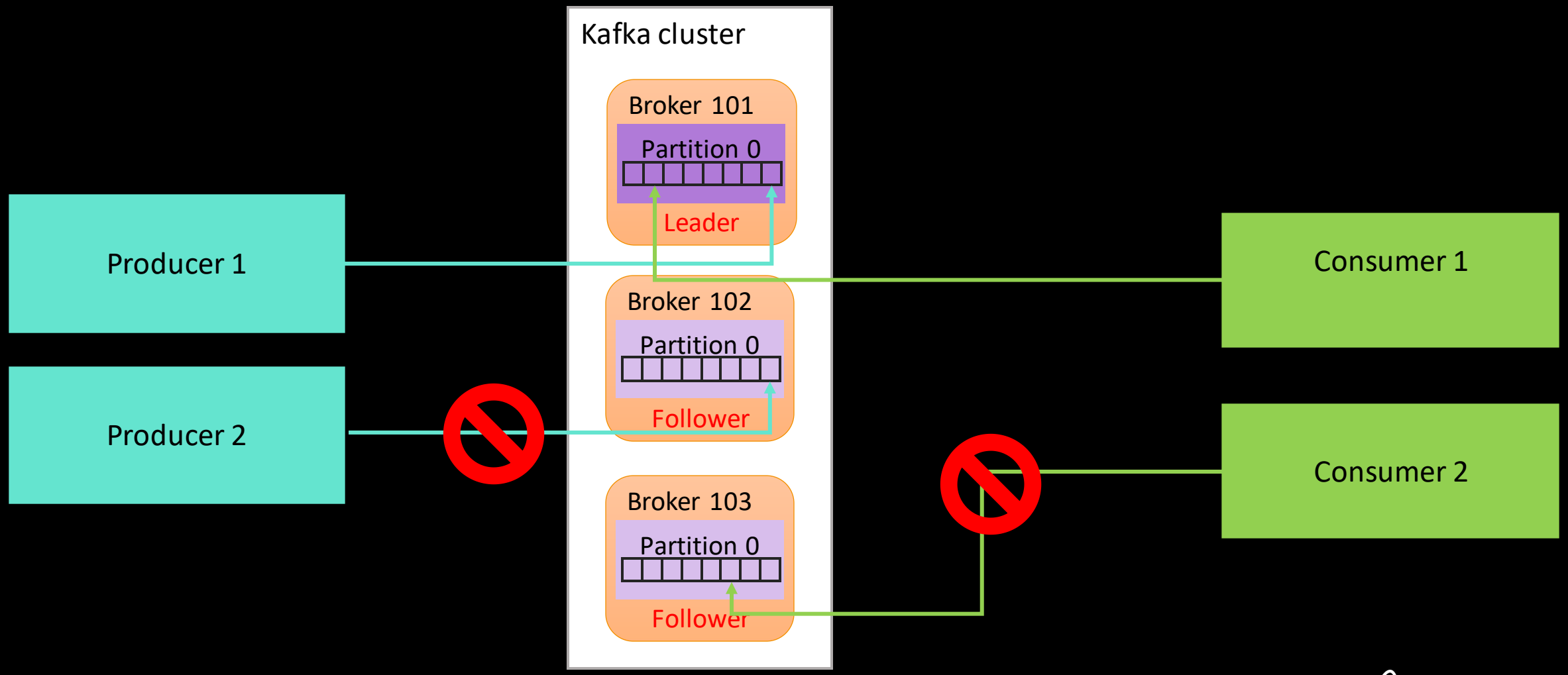
NOVATEC

Reliability & Durability: Replication of Partitions



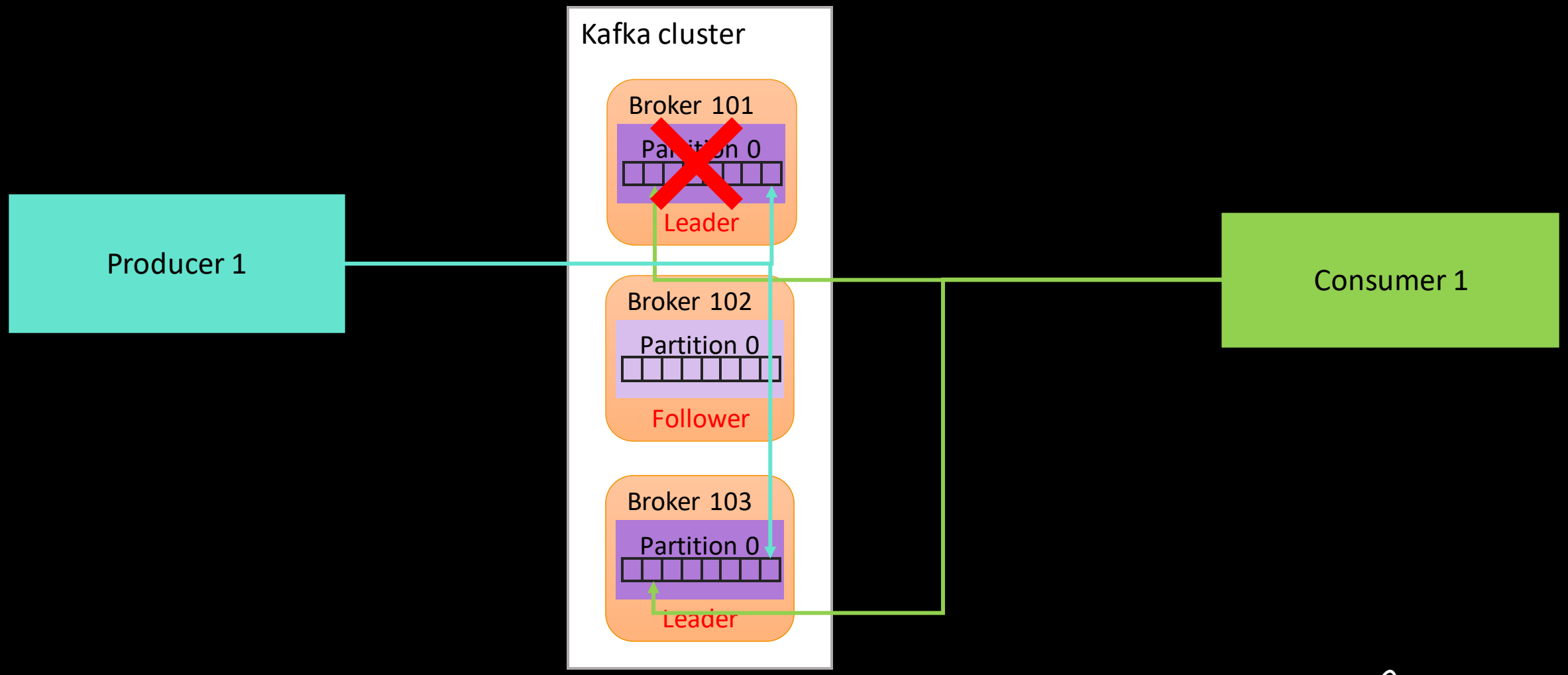
NOVATEC

Clients interact with Leaders



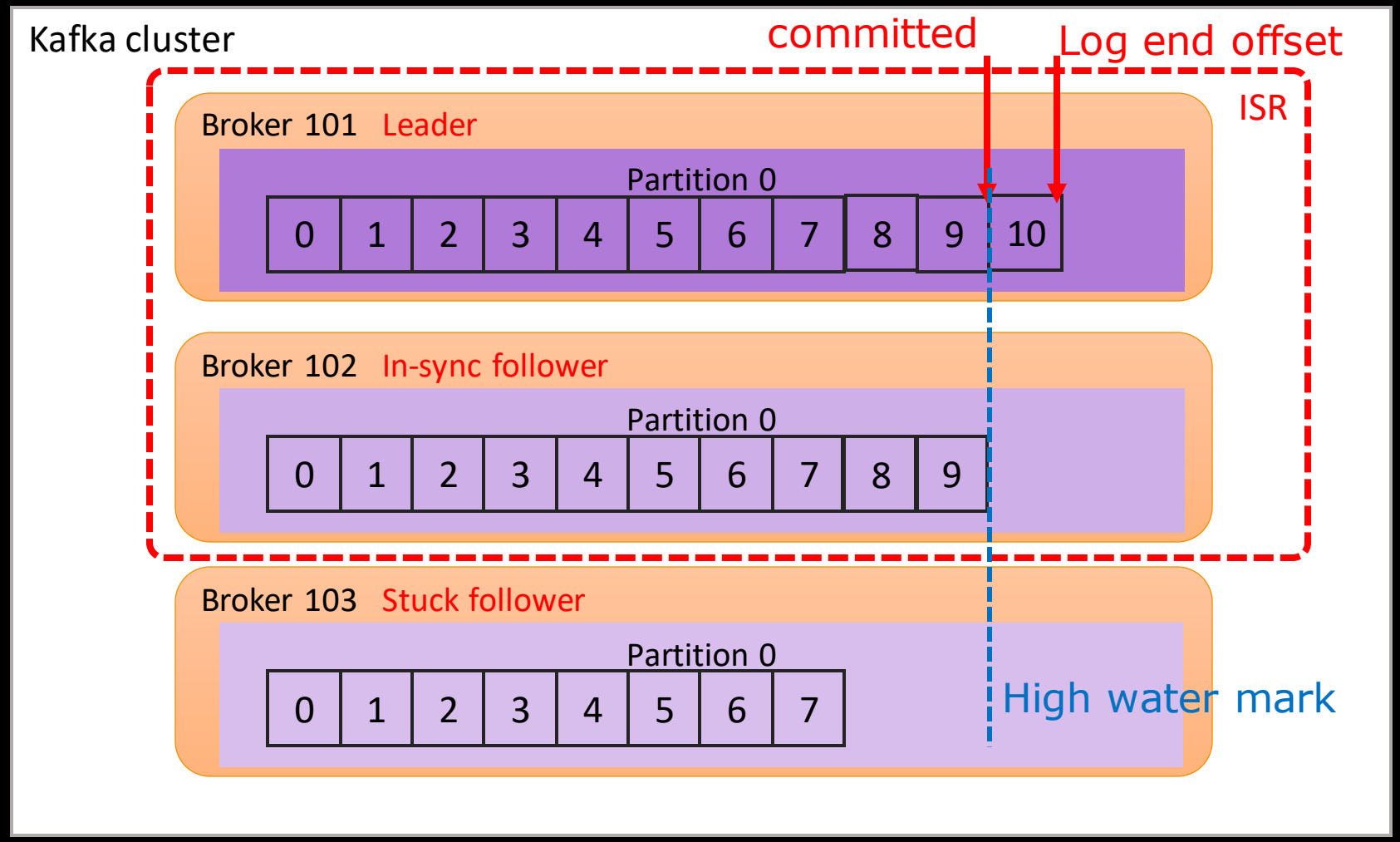
NOVATEC

Leader Failover



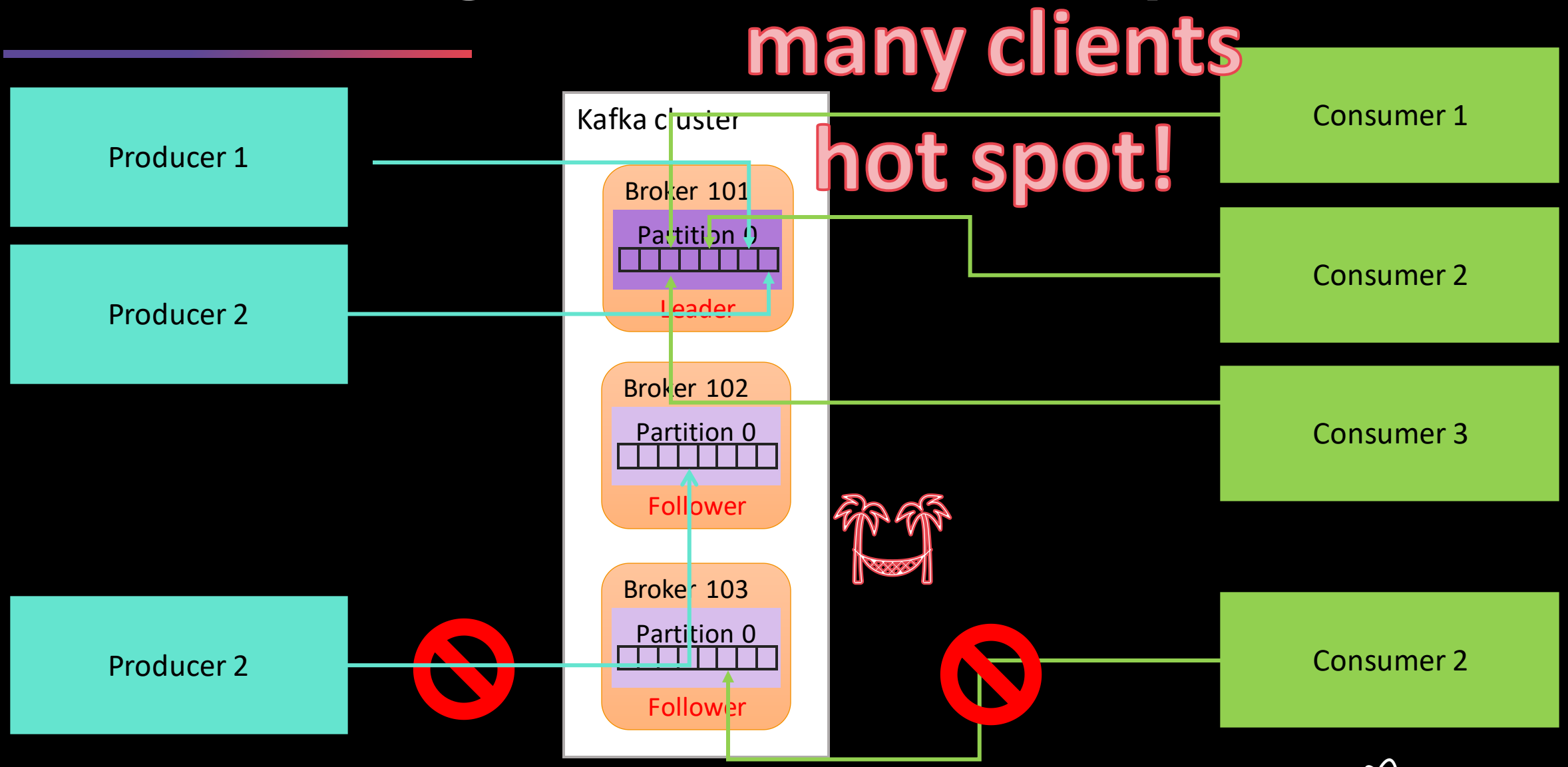
NOVATEC

In-Sync Replicas



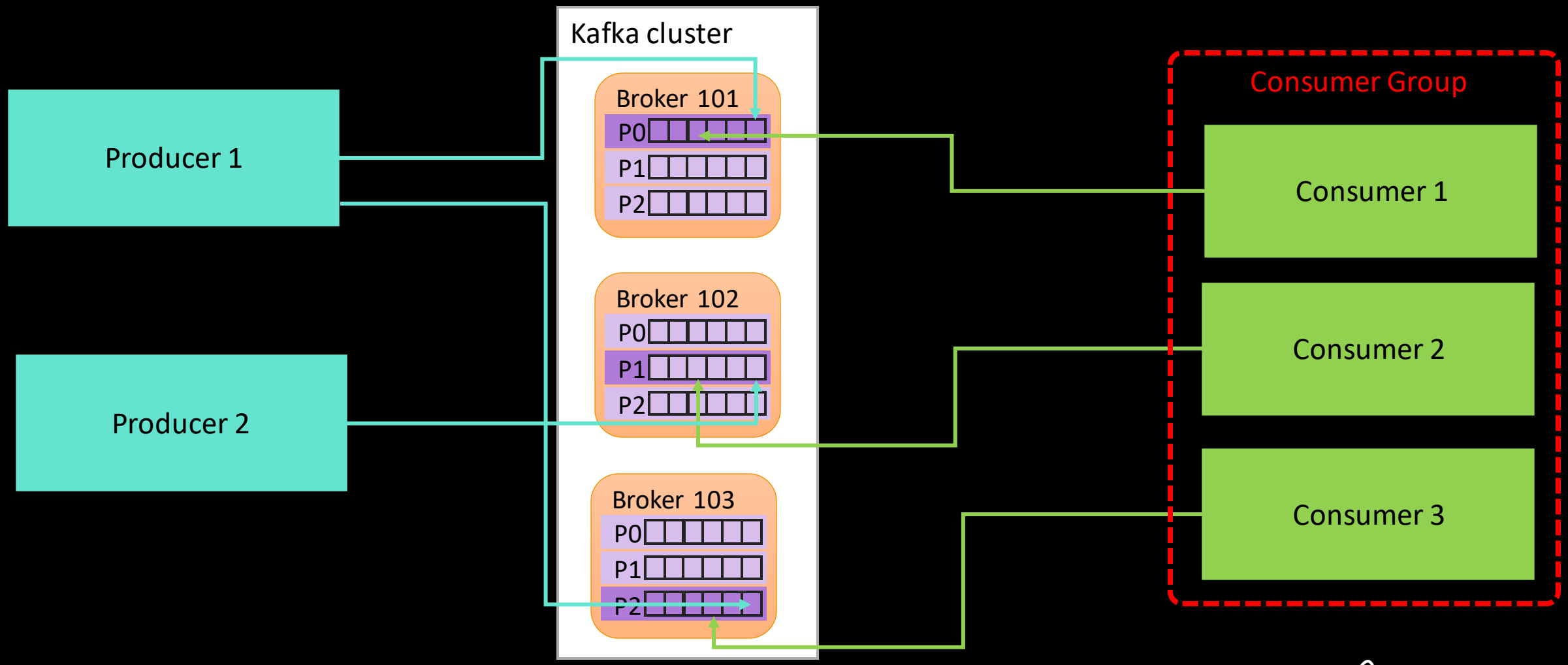
NOVATEC

Load Balancing: Partition Leadership



NOVATEC

Load Balancing Partitions Leadership (2)



NOVATEC

Zookeeper: What is it good for?

- Controller Election
- Cluster Membership
- Topic Configuration
- Quotas
- Access Control Lists

Content

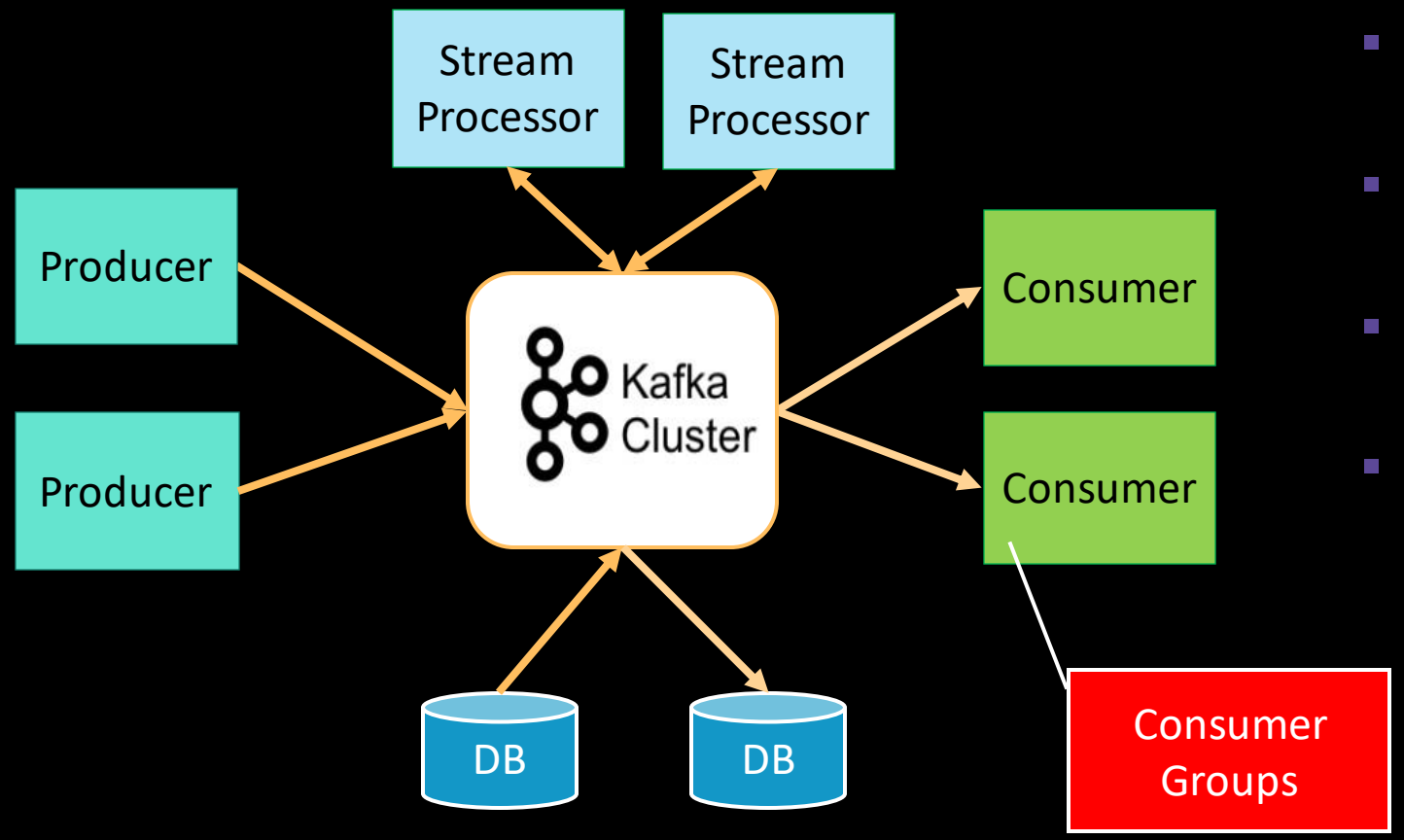
- Integration
- Event-Streaming Platform Kafka
- Asynchronous Communication
- Kafka Basics & Components
- Kafka APIs
- Kafka Examples
- Kafka Exercises

NOVATEC

Kafka APIs

NOVATEC

Kafka Core Components: 4 Core-APIs

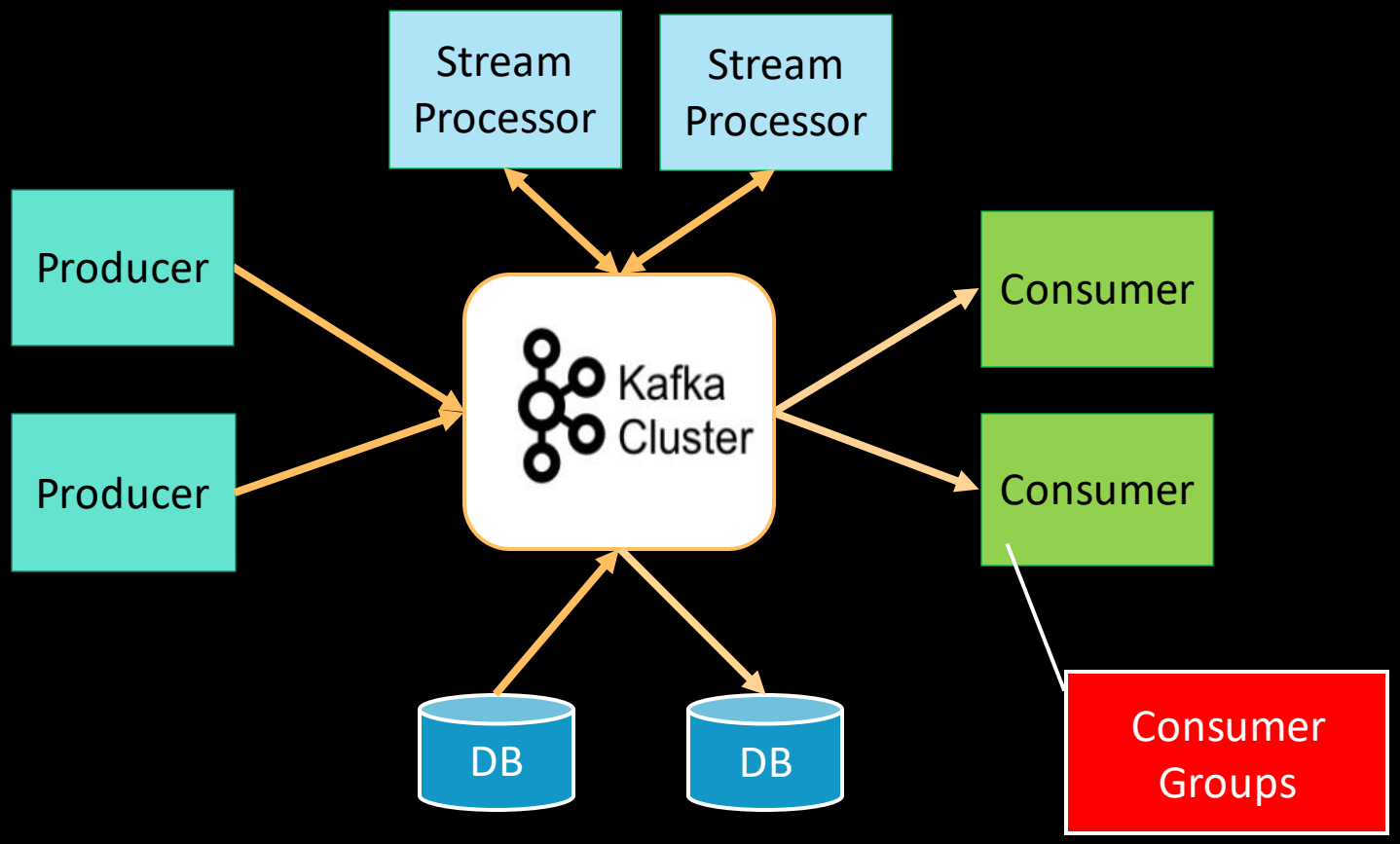


- Producer API: Enables to write messages
- Consumer API: Enables to read messages
- Streams API: Enables to analyze and transform messages
- Connect API: Enables the creation of reusable Clients

Kafka Clients supported by Confluent

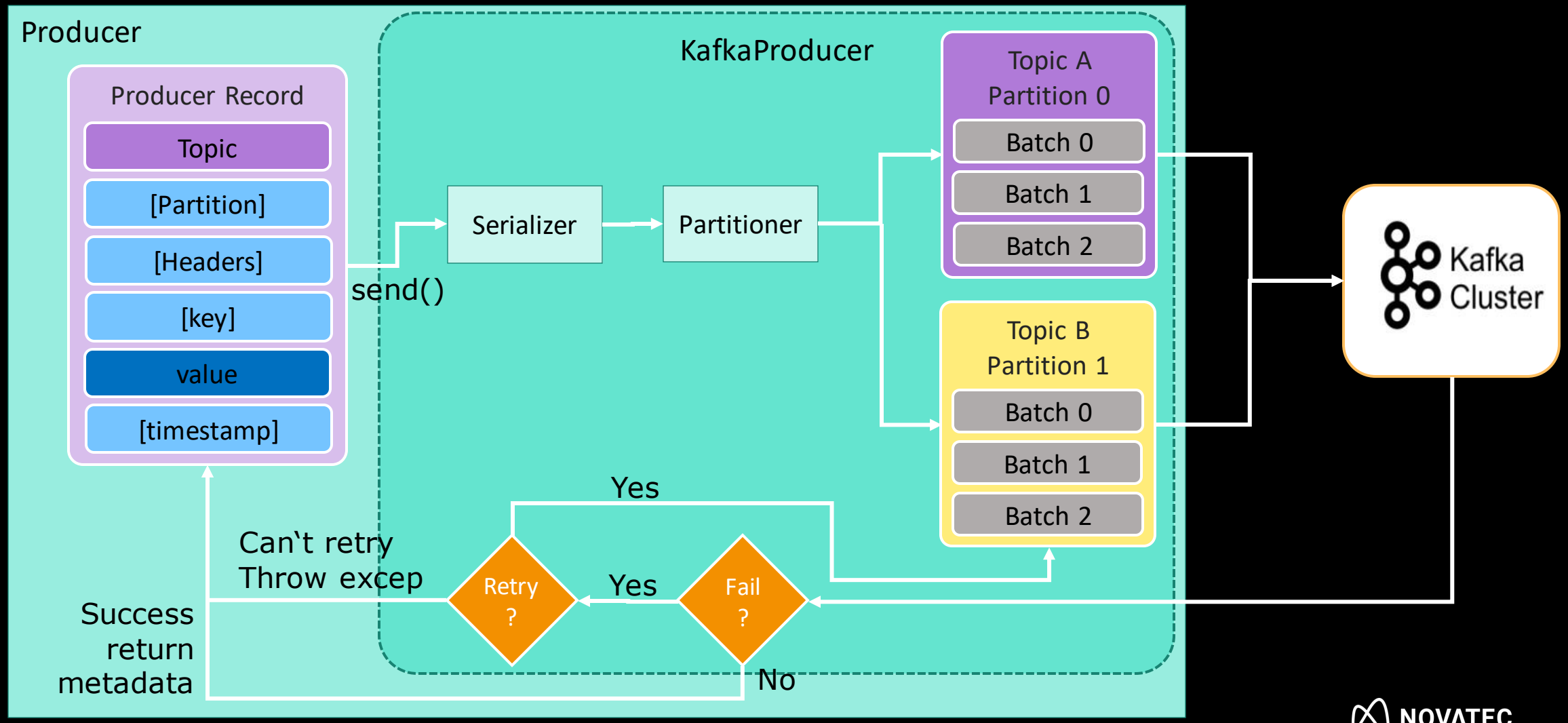
- https://github.com/confluentinc/examples/tree/5.3.1-post/clients/cloud
- client code in many languages
- JVM: Java, Groovy, Scala, Kotlin, Clojure
- C-Library: C, C#, Go, NodeJS, Python, Ruby,

Kafka Core Components: Producer API



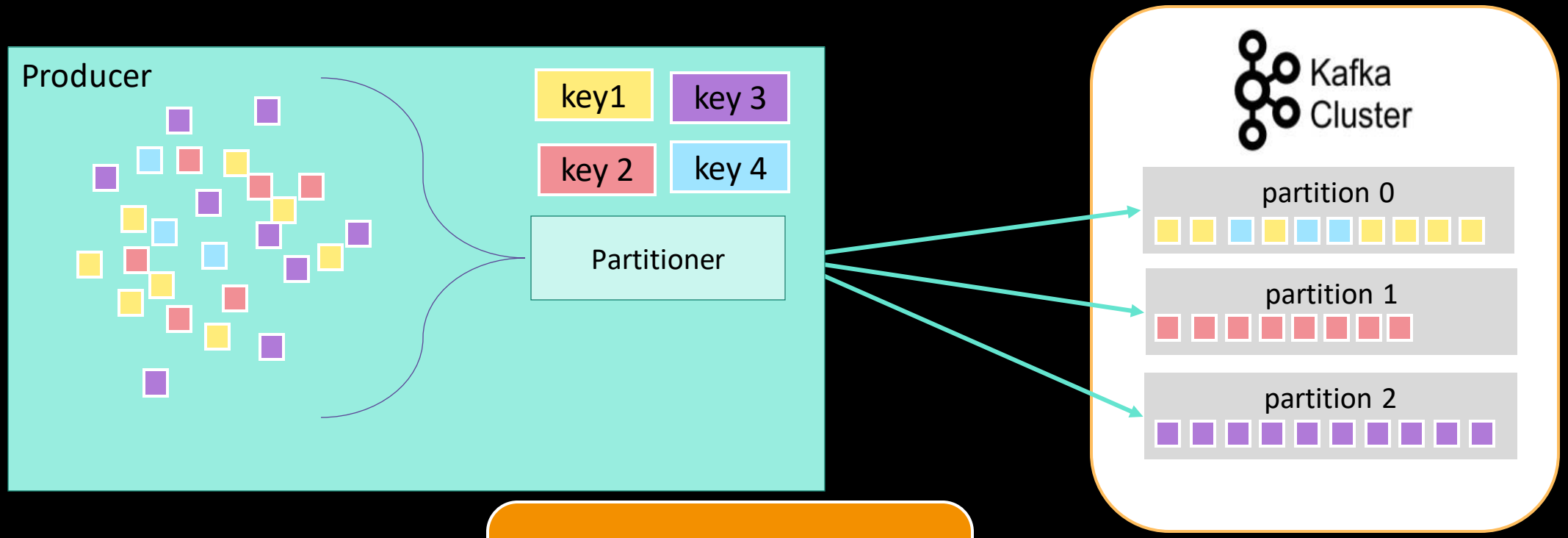
NOVATEC

Producer API: High Level Architecture & Design



NOVATEC

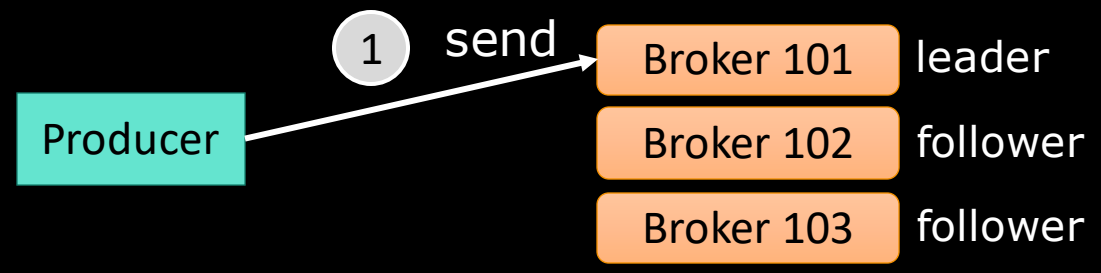
Default Partitioner



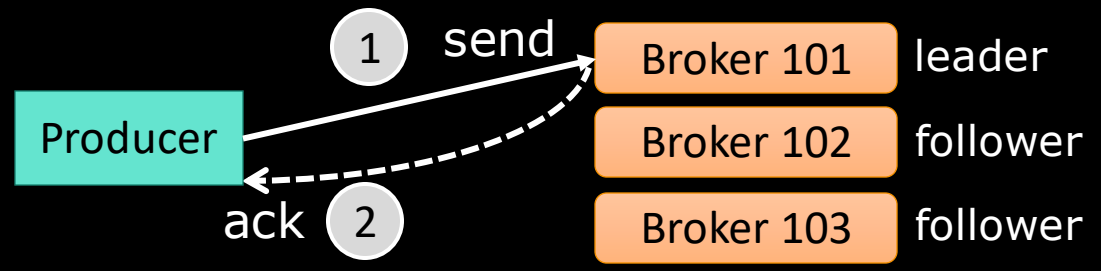
Partition = hash(key) % # partitions

NOVATEC

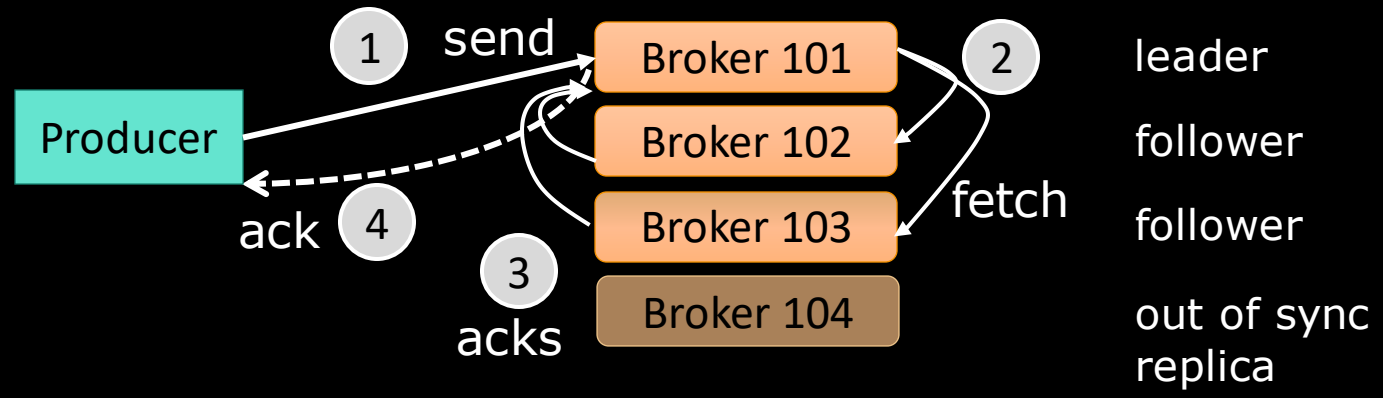
Acknowledgement



acks=0



acks=1



acks=all (-1)

NOVATEC

Development: A Basic Producer in Java

BasicProducer.java ×

```java
package clients;

import java.util.Properties;
import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.ProducerRecord;

public class BasicProducer {
    public static void main(String[] args) {
        System.out.println("*** Starting Basic Producer ***");

        Properties settings = new Properties();
        settings.put("client.id", "basic-producer-v0.1.0");
        settings.put("bootstrap.servers", "kafka-1:9092,kafka-2:9092");
        settings.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");
        settings.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");

        final KafkaProducer<String, String> producer = new KafkaProducer<>(settings);

        Runtime.getRuntime().addShutdownHook(new Thread(() -> {
            System.out.println("### Stopping Basic Producer ###");
            producer.close();
        }));

        final String topic = "hello_world_topic";
        for(int i=1; i<=5; i++){
            final String key = "key-" + i;
            final String value = "value-" + i;
            final ProducerRecord<String, String> record = new ProducerRecord<>(topic, key, value);
            producer.send(record);
        }
    }
}
```

configuration

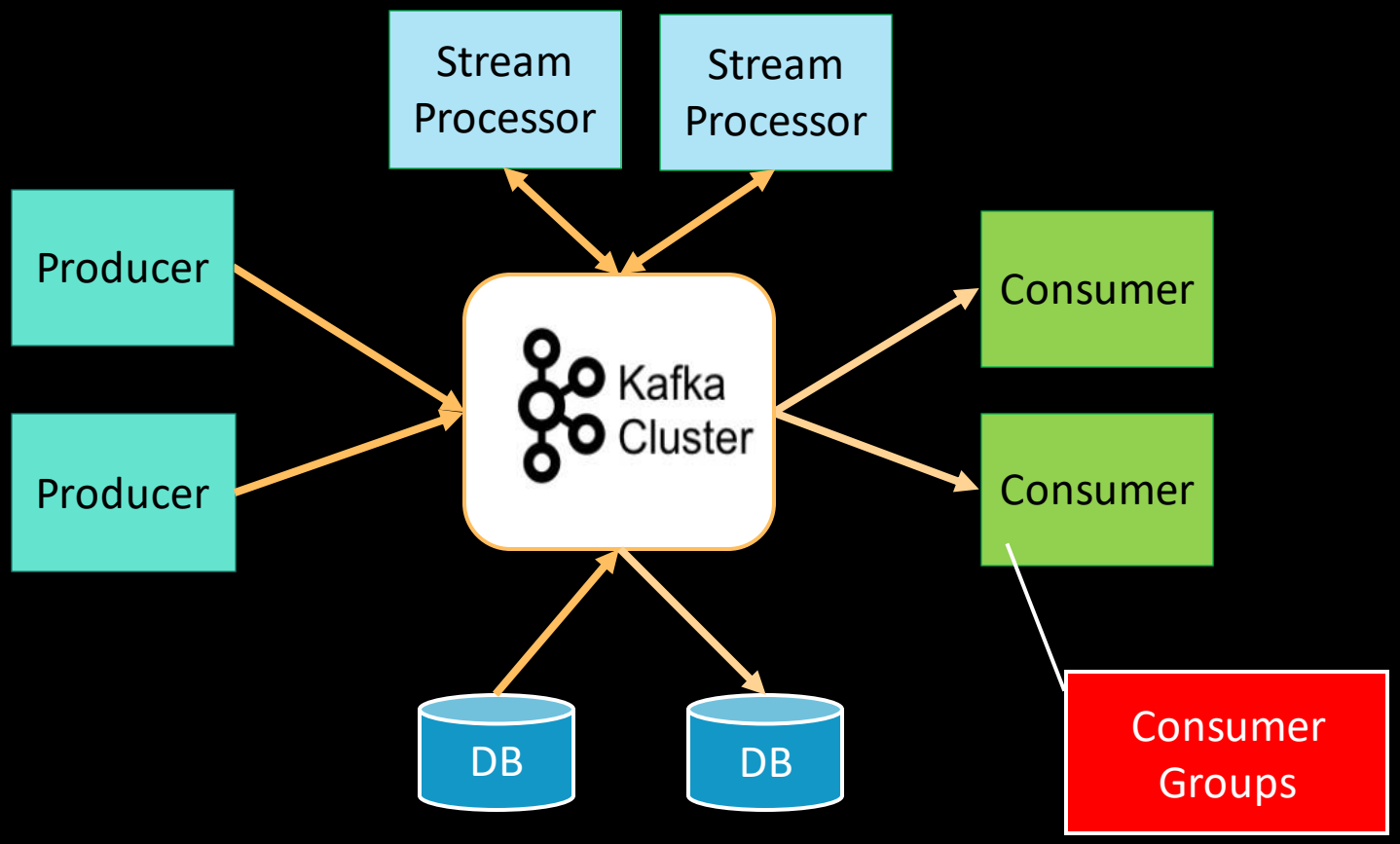
create producer

shutdown behaviour

sending data

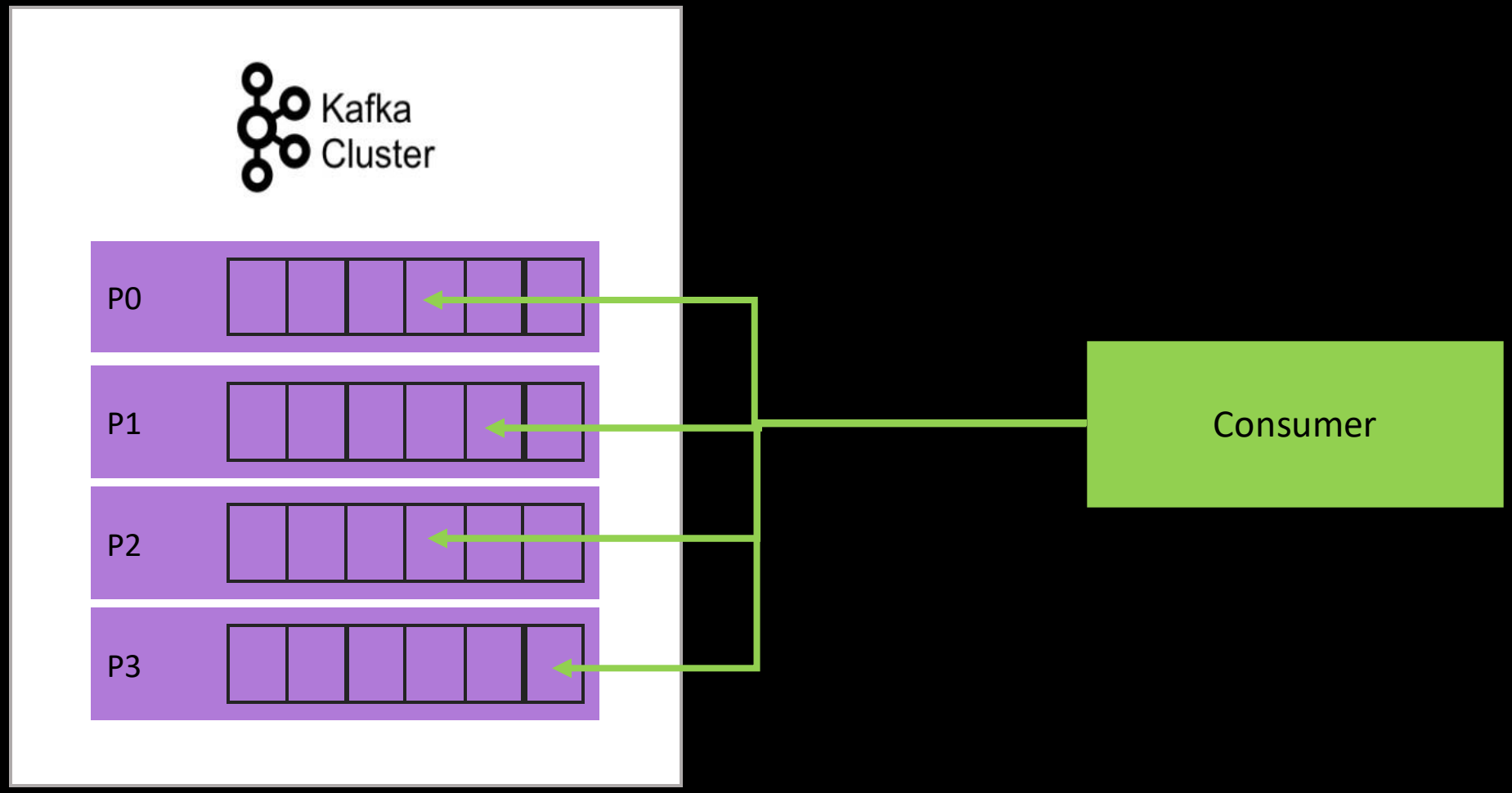
NOVATEC

Kafka Core Components: Consumer API



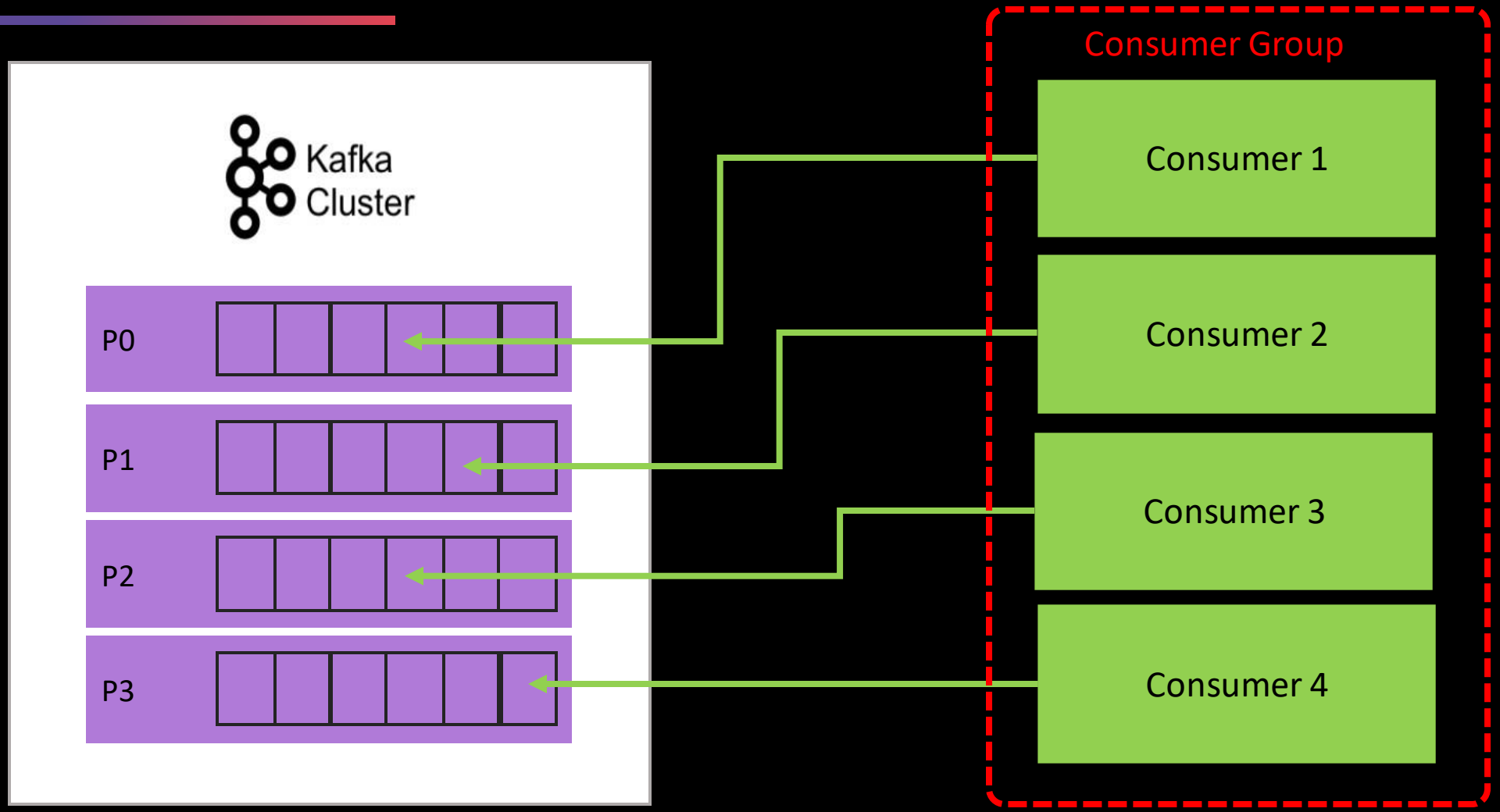
NOVATEC

Consuming from Kafka: Single Consumer



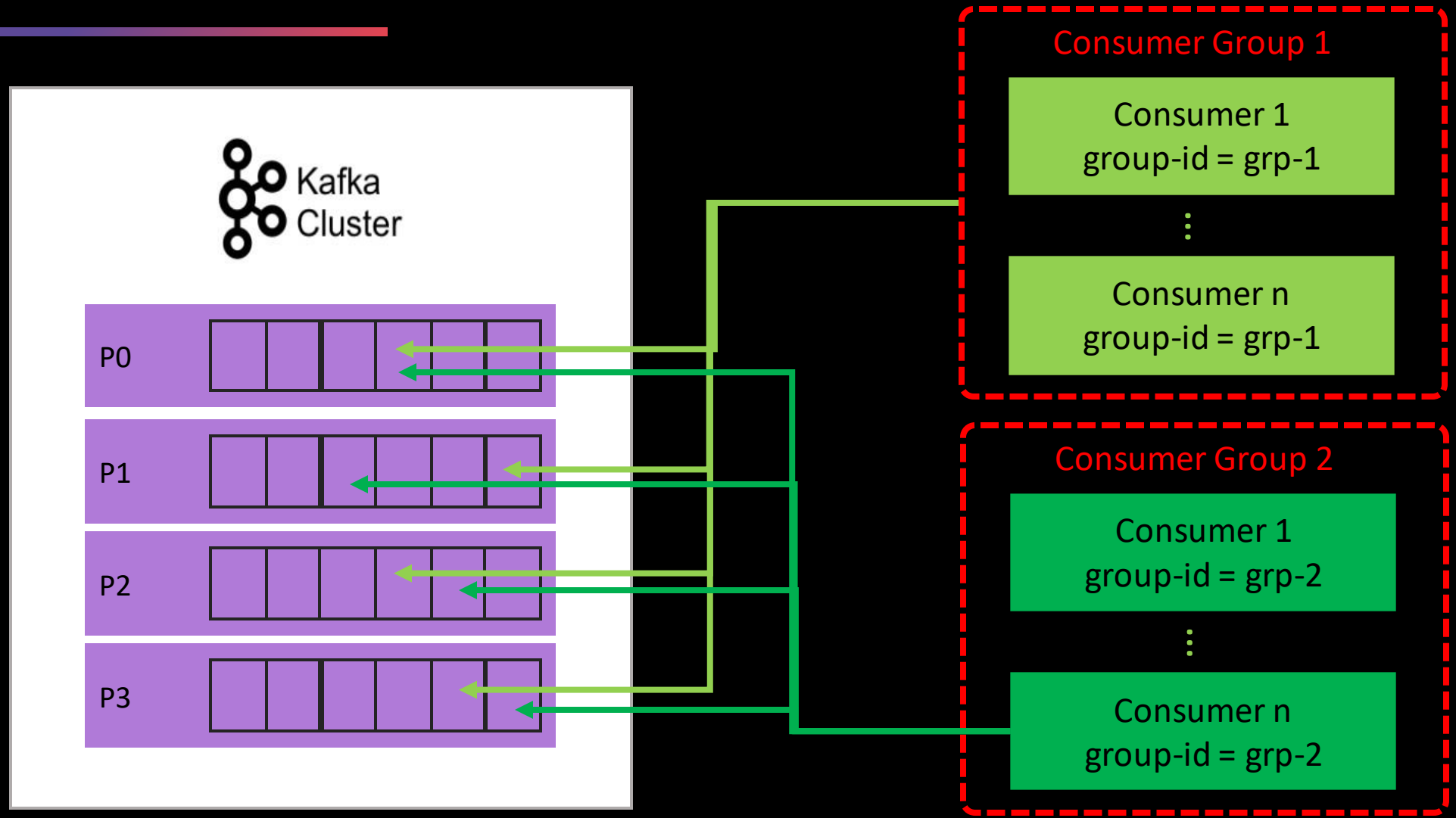
NOVATEC

Consuming from Kafka: Consumer Group



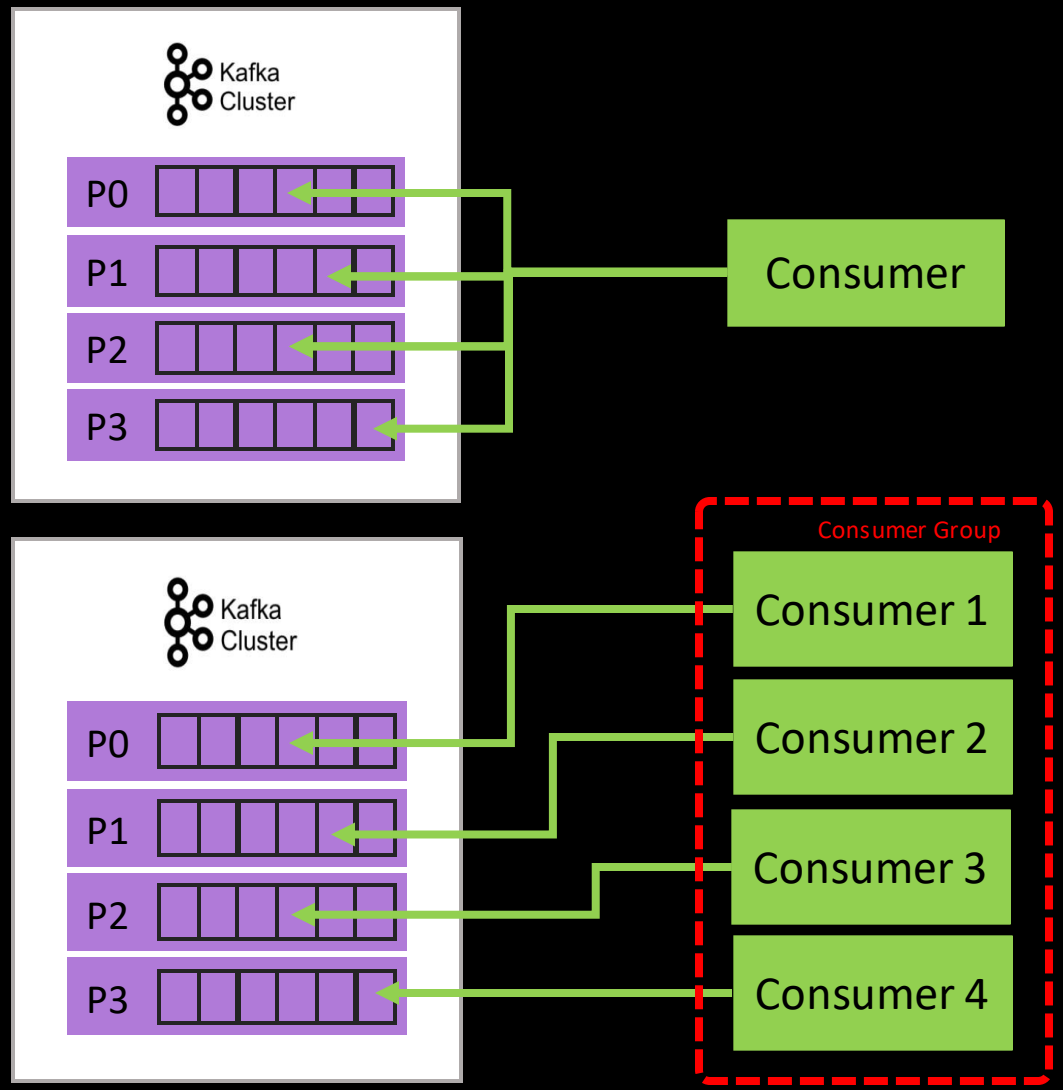
NOVATEC

Consuming from Kafka: Multiple Groups

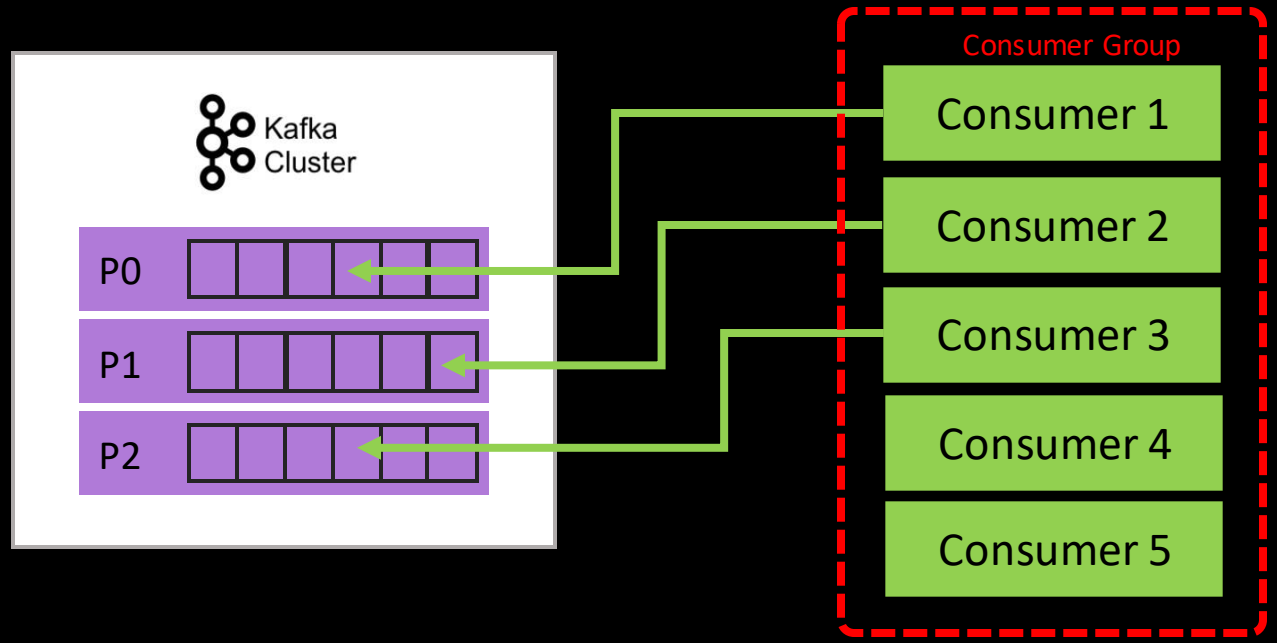


NOVATEC

Scalability is limited by Number of Partitions

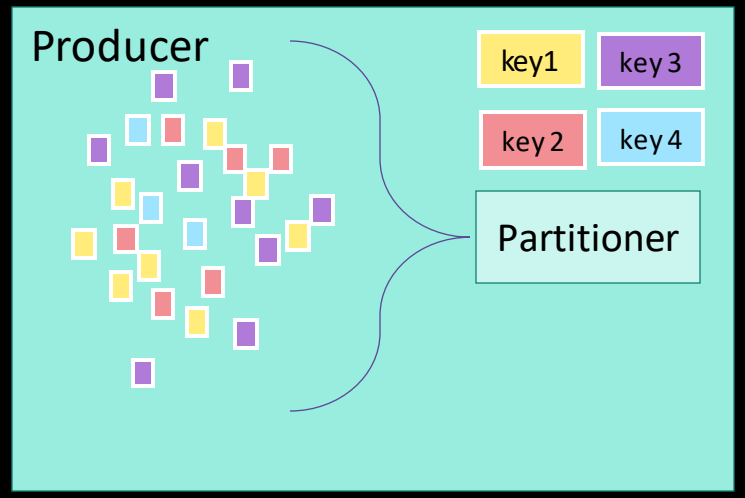


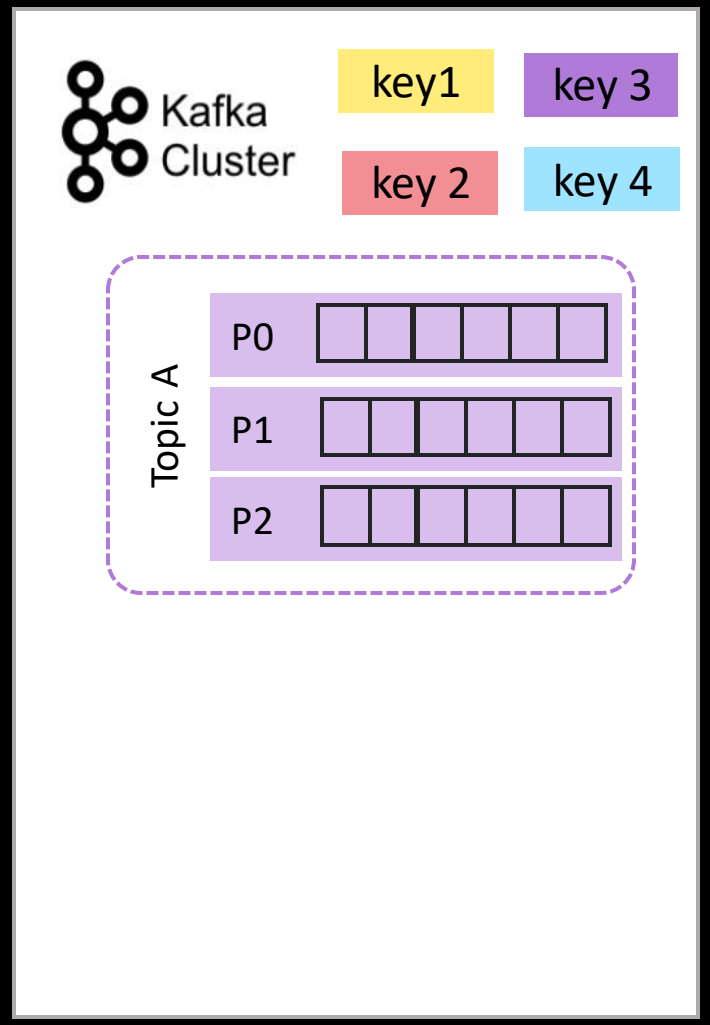
- number of useful consumers in a group is constrained by the number of partitions

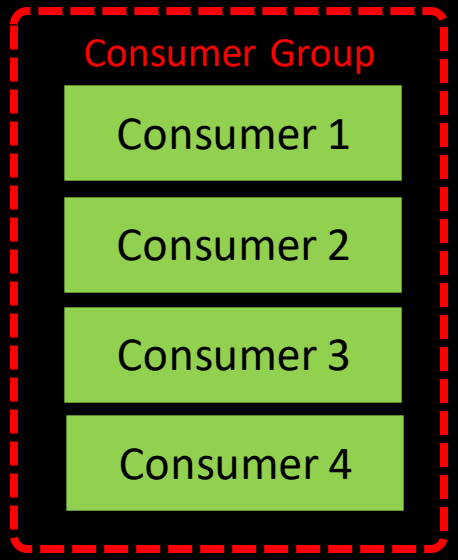


NOVATEC

How are Partitions assigned to Consumers 1

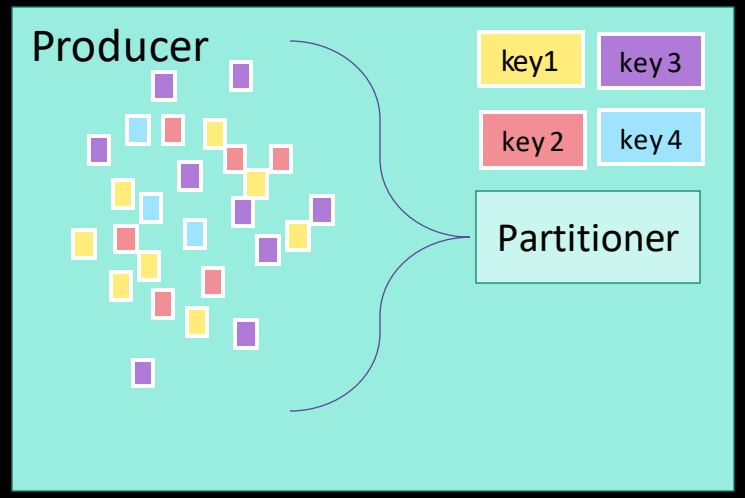


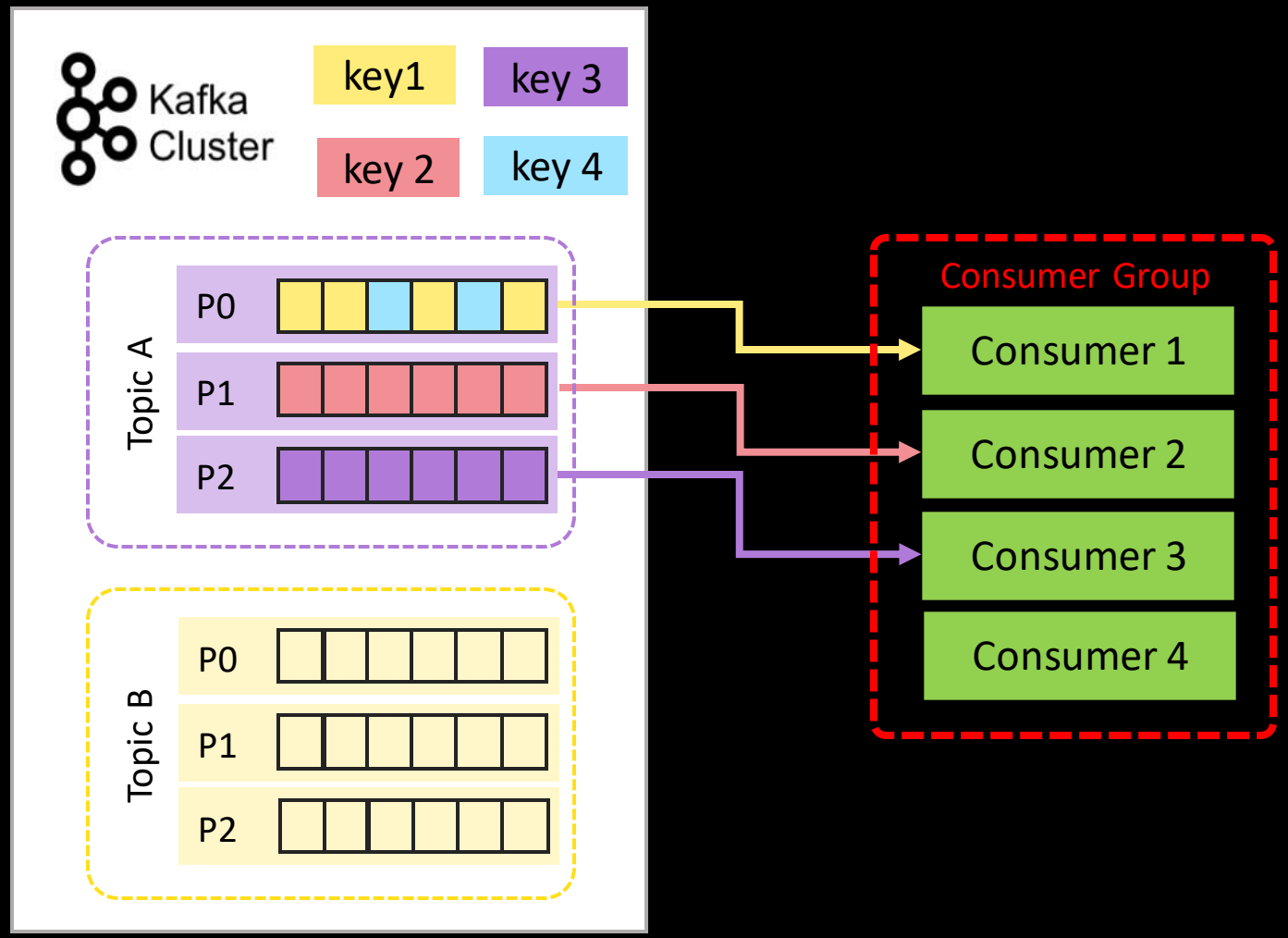




NOVATEC

How are Partitions assigned to Consumers 2

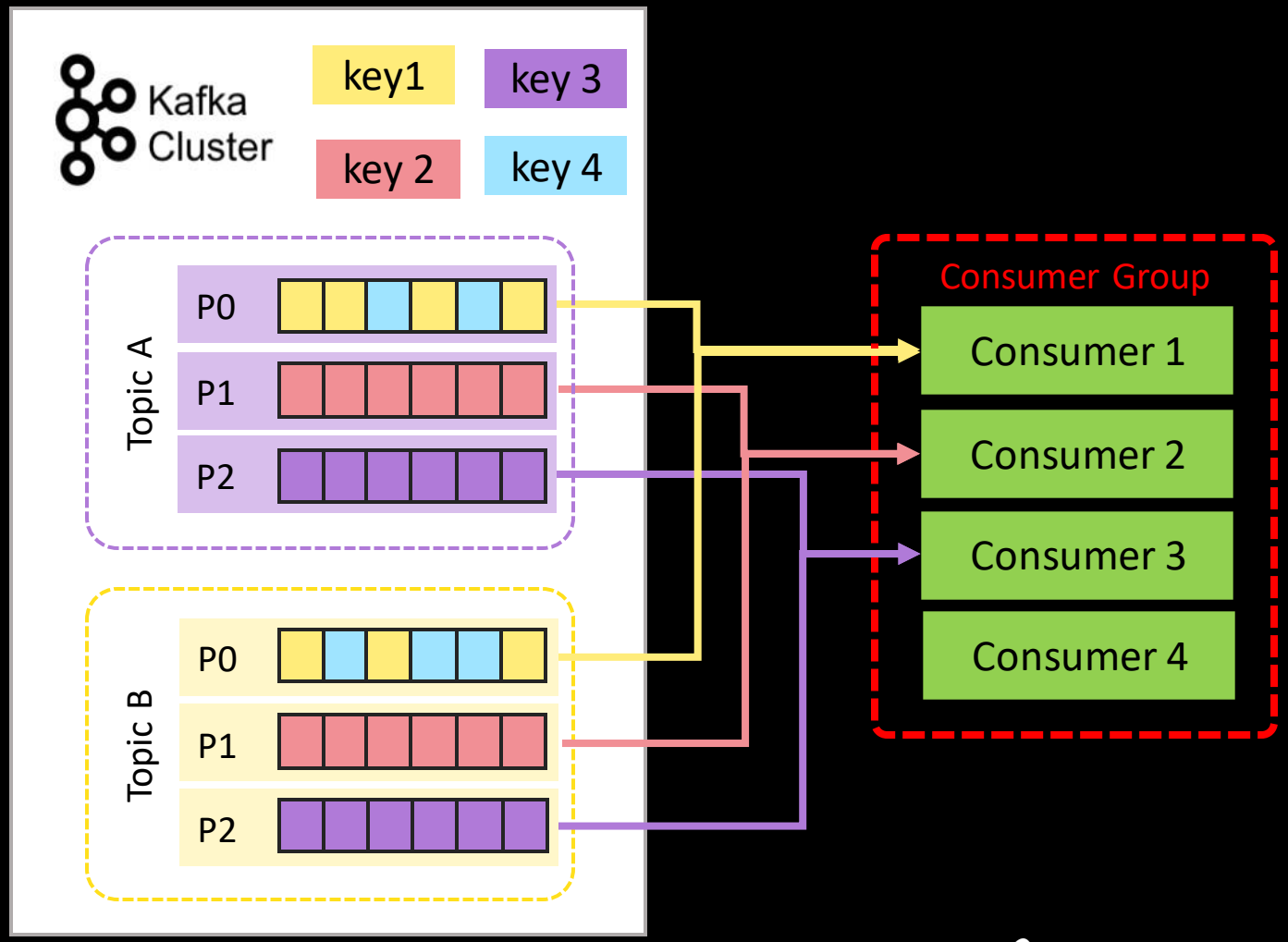




NOVATEC

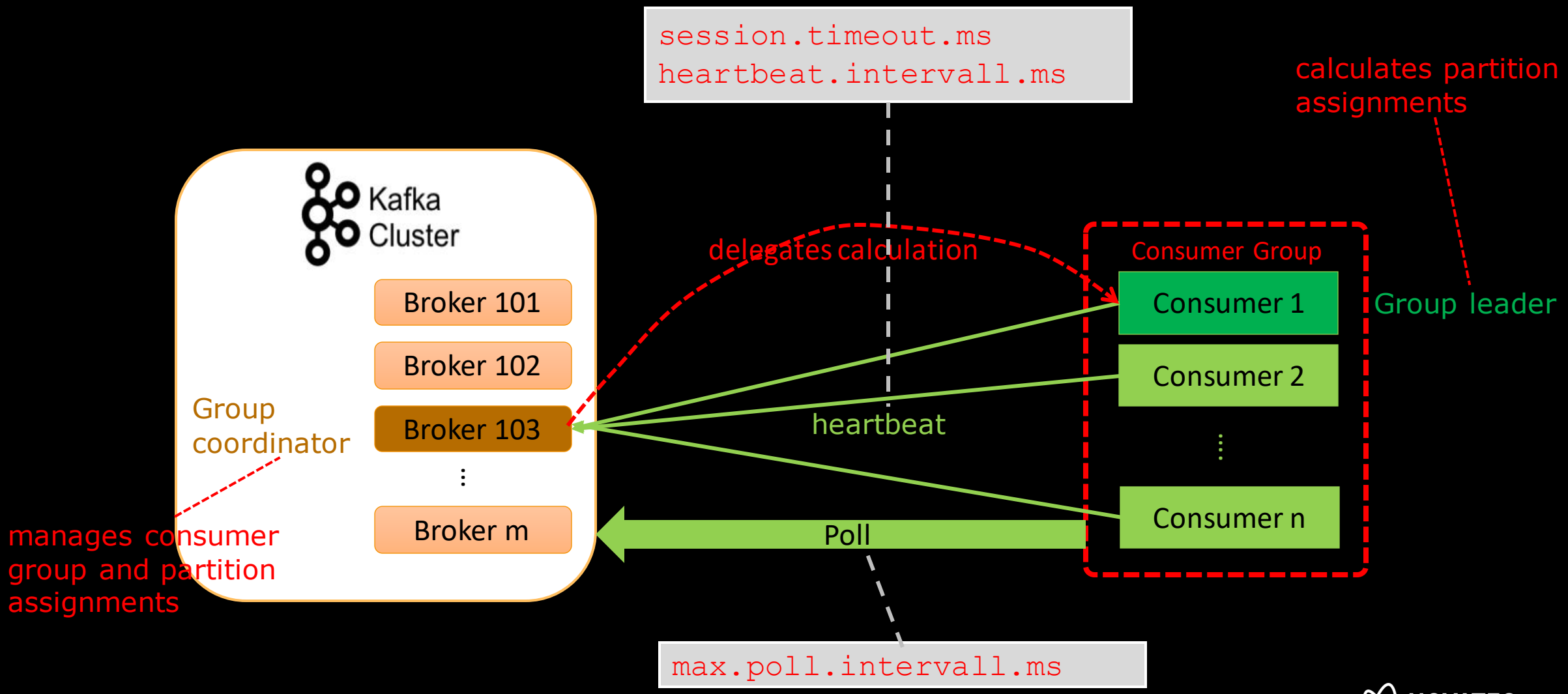
How are Partitions assigned to Consumers 3

- partition.assignment.strategy consumer property:
  - RangeAssignor (used in stream-processing for co-partitioned topics)
  - RoundRobinAssignor
  - StickyAssignor



NOVATEC

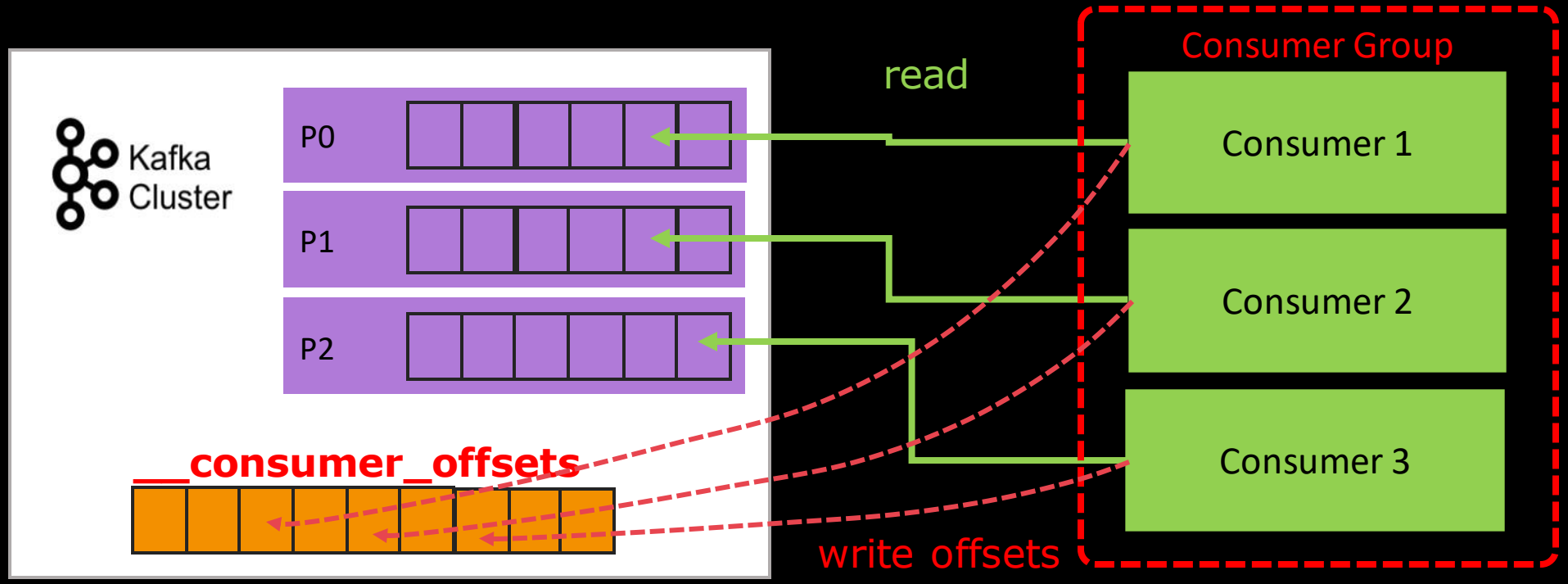
Consumer Liveliness 1-3



NOVATEC

Consumers and Offsets

- Offset: Position of a record in the partition
- Group_id, topic, partition is tracked in topic: __consumer_offsets
- Consumer Offset Topic tracks which message should be read next



NOVATEC

Development: Basic Consumer in .NET/C#

```
8   namespace consumer_net {
        0 references
9       class Program {
            0 references
10          static void Main (string[] args) {
11              Console.WriteLine ("Starting Consumer!");
12              var config = new Dictionary<string, object> {
13                  { "group.id", "dotnet-consumer-group" },
14                  { "bootstrap.servers", "kafka-1:9092" },
15                  { "auto.commit.interval.ms", 5000 },
16                  { "auto.offset.reset", "earliest" }
17              };
18
19              var deserializer = new StringDeserializer (Encoding.UTF8);
20              using (var consumer = new Consumer<string, string> (config, deserializer, deserializer)) {
21                  consumer.OnMessage += (_, msg) =>
22                      Console.WriteLine ($"Read ('{msg.Key}', '{msg.Value}') from: {msg.TopicPartitionOffset}");
23
24                  consumer.OnError += (_, error) =>
25                      Console.WriteLine ($"Error: {error}");
26
27                  consumer.OnConsumeError += (_, msg) =>
28                      Console.WriteLine ($"Consume error ({msg.TopicPartitionOffset}): {msg.Error}");
29
30                  consumer.Subscribe ("hello_world_topic");
31
32                  while (true) {
33                      consumer.Poll (TimeSpan.FromMilliseconds (100));
34                  }
35              }
36          }
37      }
38  }
```

configuration

subscribing to topic

message handling

error handling

polling data

NOVATEC

Content

- Integration
- Event-Streaming Platform Kafka
- Asynchronous Communication
- Kafka Basics & Components
- Kafka APIs
- Kafka Examples
- Kafka Exercises

NOVATEC

Kafka Examples

Bosch Powertools & Deutsche Bahn Passenger Information

NOVATEC

Todays Ecosystems are pretty big

- 2.2 X 10^12 messages per day (6 x 10^15 Byte Petabyte)
- up to 400 Microservices per cluster
- 20 – 200 Broker per cluster

NETFLIX

BOSCH

EN

Projects > Neubau Klinik > Tasks

Create Assign Send

10 20 50

1 2 3 4 5 ... 128

1-10/1279

Task Title	Location	Start	Due	Craft	Company	State	News
4.OG Technikzentrale, Stahlbau	-	07/07/20	07/14/20	Rohbauarbeiten	Company B1 Anne Esch		
Abbau Kran klein	-	09/12/20	09/12/20	Rohbauarbeiten	Company B1 Anne Esch		
Abbruch Kanal in Baufeld	-	09/11/19	11/09/19	Erdbauarbeiten	Company B1 Anne Esch		
Abbruch Verbindungsgang	-	08/06/19	08/17/19	Erdbauarbeiten	Company B1 Anne Esch		
Abdichtung	-	01/11/21	01/16/21	Fliesen	Assign now		
Abdichtung	-	01/25/21	01/30/21	Fliesen	Assign now		
Abdichtung	-	12/28/20	01/02/21	Fliesen	Assign now		
Abdichtung + Dämmung Achse P-O...	-	05/13/20	05/14/20	Rohbauarbeiten	Company B1		

RefinemySite: Lean Construction SaaS

Kafka as Single Source of Truth

Confluent Cloud

Kafka as Event-Sourcing Backbone

Replication of Data between Microservices via Topics

Simple Bootstrapping of new Services by reprocessing the Event Stream

Real-Time & Near Realtime Notifications of Users

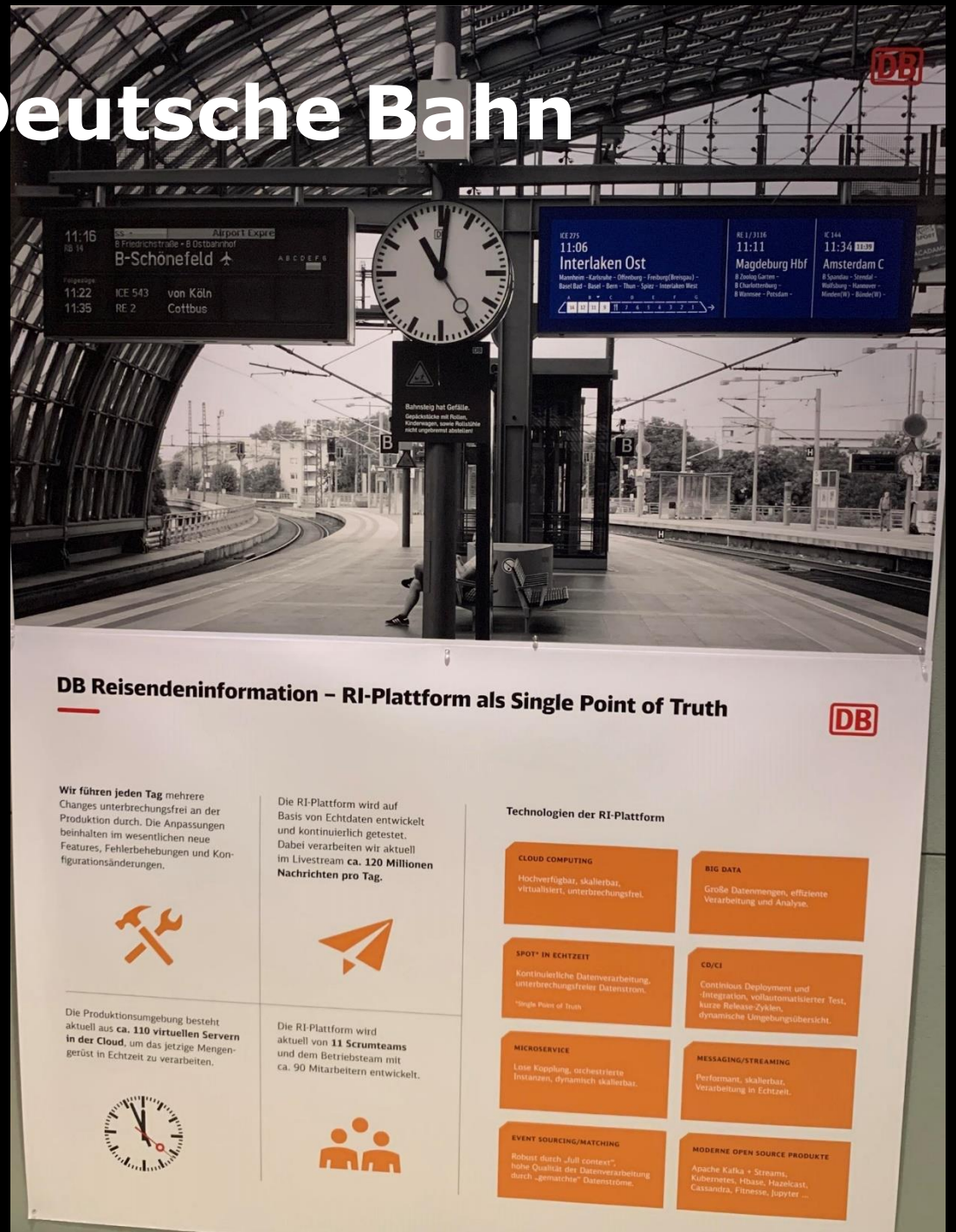
19 independently deployable Units

29 People across Europe, started in 2016

Topics with different Avro-Schemas

Passenger Information of Deutsche Bahn

- Talk at Confluent Streaming event 11.11.2019



Kafka-Project: DB Passenger Information

Facts and Figures

- ~100 persons in 12 Scrum Teams
- 24/7 running (DevOps)
- multiple daily deployments in production
- ~100 virtual servers
- ~100 Microservices

Kafka

Broker	Msg In	Volume In
6 in 3 AZs	**300M /day**	**1TB /day**

Topic/ Partition	Msg In (Avg)	Volume In
320 /3,3k	**4k /sec**	**13MB /sec**

February 2020 (OOP Talk DB/Novatec)

NOVATEC

Content

- Integration
- Event-Streaming Platform Kafka
- Asynchronous Communication
- Kafka Basics & Components
- Kafka APIs
- Kafka Examples
- Kafka Exercises

NOVATEC

Kafka Exercises

Introduction and preparation of the next unit

NOVATEC

Sources

NOVATEC

Sources

1. https://www.enterpriseintegrationpatterns.com/patterns/messaging/
2. Enterprise Integration Patterns, Gregor Hohpe and Bobby Woolf: ISBN 0321200683
3. https://kafka.apache.org/
4. https://www.confluent.io/what-is-apache-kafka/
5. https://www.confluent.io/resources/
6. https://www.informatik-aktuell.de/betrieb/verfuegbarkeit/apache-kafka-eine-schluesselplattform-fuer-hochskalierbare-systeme.html
7. https://www.slideshare.net/KaiWaehner/apache-kafka-vs-integration-middleware-mq-etl-esb?ref=https://www.kai-waehner.de/blog/2019/03/07/apache-kafka-middleware-mq-etl-esb-comparison/
8. https://www.confluent.io/blog/apache-kafka-vs-enterprise-service-bus-esb-friends-enemies-or-frenemies/
9. https://microservices.io

Christopher Uldack
Senior Consultant

Christopher.Uldack@novatec-gmbh.de

Oliver Berger
Senior Managing Consultant

Oliver.Berger@novatec-gmbh.de

Novatec Consulting GmbH
Dieselstraße 18/1
D-70771 Leinfelden-Echterdingen

T. +49 711 22040-700
info@novatec-gmbh.de
www.novatec-gmbh.de