Apache Kafka.

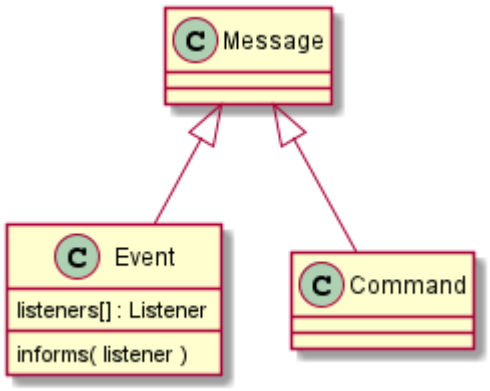
1. Kafka

1.1. Event-driven architecture:

- Microservices
- Serverless
- FaaS
- Streaming
- Event Sourcing
- CQRS

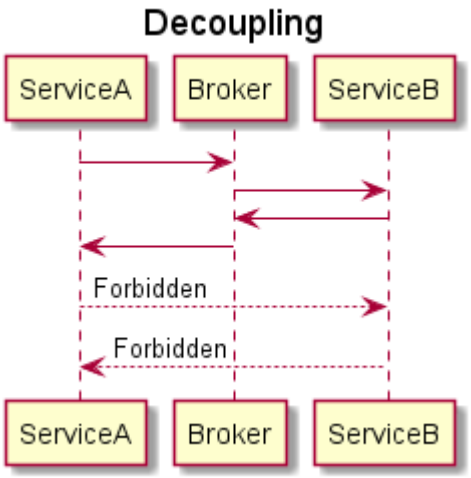
1.2. Event-driven architecture basic units

- Message - is a basic unit of commutication(string, number, ...) it is described as a generic interface which has no special intent
- Event - is a message, it informs various listeners about something that has happend
- Command - targeted action by having one to one connection between producer and comsumer(ordering something can be viewed as a command)

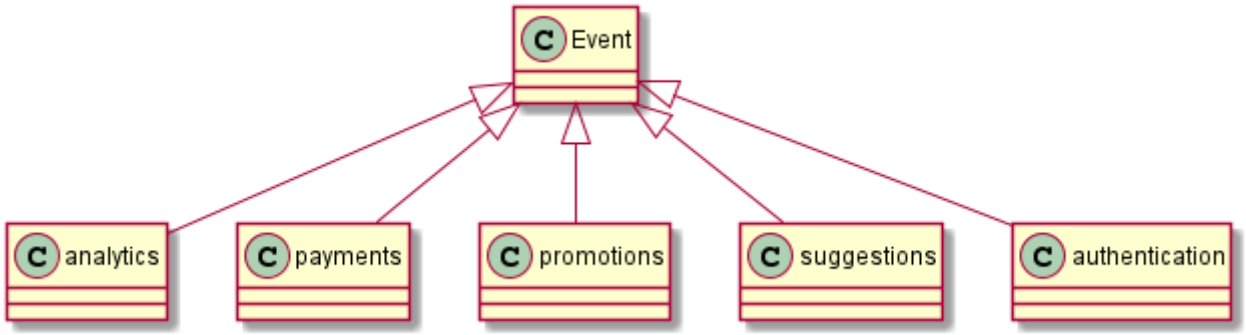


1.3. Benefits of EDA

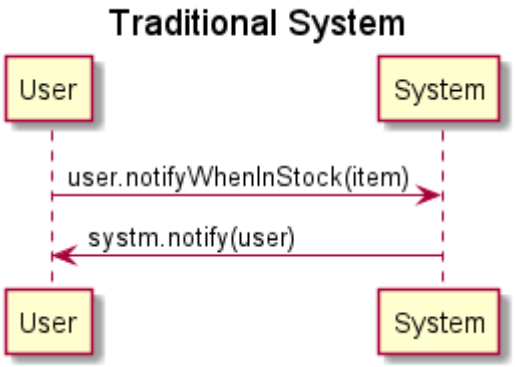
- Decoupling - all the components are decoupled one of the other, just a classes in a programming languages.

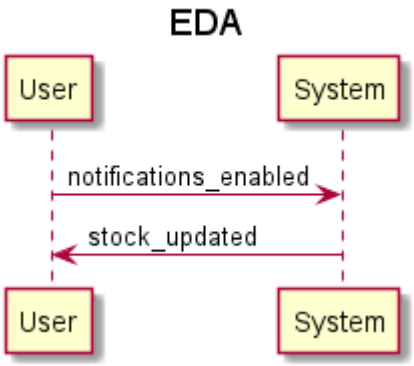


- Encapsulation - Events can be classified within different functional boundaries and they can be processed on the same boundaries(for example in ecommerse system we would have events related to analytics, payments, promotions, suggestions, authentication. There are clear boundaries between each type of event without having to worry about confusing them).

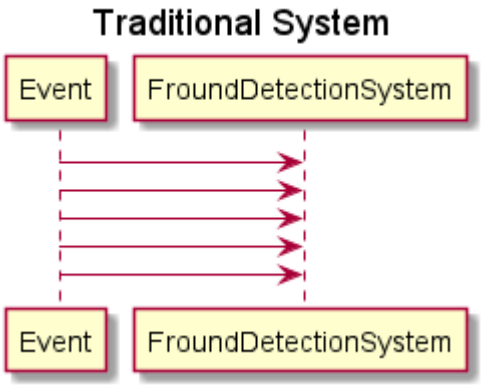


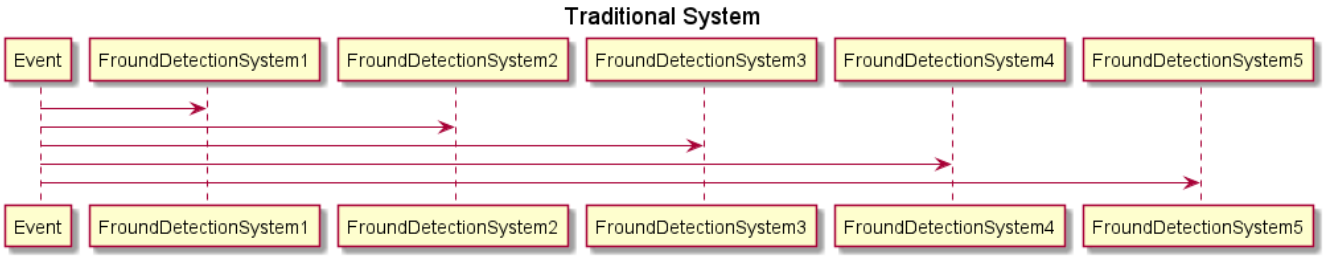
- Optimization





- Scalability





1.4. EDA Drawbacks

- Steap learning curve
- Complexity
- Loss of transactionality
- Lineage (debugging system, events can be lost or corrupted, because applications ar losely coupled)

2. Apache Kafka

2.1. Windows

2.1.1. Install

Downlaod

```
tar -xzf kafka_2.13-2.6.0.tgz
cd kafka_2.13-2.6.0
```

2.1.2. Run

```
D:\kafka_2.13-2.6.0\bin\windows\zookeeper-server-start.bat D:\kafka_2.13-
2.6.0\config\zookeeper.properties
D:\kafka_2.13-2.6.0\bin\windows\kafka-server-start.bat D:\kafka_2.13-
2.6.0\config\server.properties
```

2.2. Confluent

quick start

/root/confluent-6.0.0/etc/kafka/server.properties

```
# Uncomment this property to allow remote access
advertised.listeners=PLAINTEXT://192.168.56.10:9092
```

confluent local services

```
confluent local services start
# The output is "Error: ZooKeeper failed to start". Despite this run next commands
confluent local services kafka-rest start
confluent local services kafka start
confluent local services control-center start
http://192.168.56.10:9021/
```

```
confluent local services connect start && confluent local services kafka start &&
confluent local services kafka-rest start && confluent local services ksql-server
start
http://192.168.56.10:9021/
```

```
confluent local services ksql-server start
/root/confluent-6.0.0/bin/ksql-server-start /root/confluent-6.0.0/etc/ksqldb/ksql-
server.properties
/root/confluent-6.0.0/bin/ksql
```

```
confluent local services list start
confluent local services schema-registry start
```

schema-registry-ui

```
git clone https://github.com/Landoop/schema-registry-ui.git
cd schema-registry-ui
npm install
npm start
http://192.168.56.10:8080
```

```
cd D:\kafka_2.13-2.6.0\bin\windows
kafka-topics --create --topic quickstart-events --bootstrap-server 192.168.56.10:9092
kafka-topics --describe --topic quickstart-events --bootstrap-server
192.168.56.10:9092
kafka-console-producer --topic quickstart-events --bootstrap-server 192.168.56.10:9092
kafka-console-consumer --topic quickstart-events --from-beginning --bootstrap-server
192.168.56.10:9092
kafka-console-consumer --topic user-tracking --from-beginning --bootstrap-server
192.168.56.10:9092
```

2.3. KAFDROP UI

```
git clone https://github.com/obsidiandynamics/kafdrop.git
cd kafdrop
mvn clean package -DskipTests
java --add-opens=java.base/sun.nio.ch=ALL-UNNAMED -jar target/kafdrop-3.28.0-
SNAPSHOT.jar --kafka.brokerConnect=localhost:9092
http://192.168.56.10:9000/
```

3. Schema registry

```
git clone https://github.com/confluentinc/schema-registry.git
cd schema-registry/
git checkout v5.2.0
mvn clean package -DskipTests
```

simpliest schema registry

```
git clone https://github.com/Landoop/schema-registry-ui.git
cd schema-registry-ui
npm install
npm start
http://localhost:8080
```

3.1. Linux

TODO

3.2. Kafka Tool

Kafka Tool

4. Topics

```
D:\kafka_2.13-2.6.0\bin\windows\kafka-topics.bat --create --bootstrap-server
localhost:9093 --partitions 2 --replication-factor 2 --topic user-tracking

D:\kafka_2.13-2.6.0\bin\windows\kafka-topics.bat --list --bootstrap-server
localhost:9093 user-tracking
```

5. Avro

5.1. Compile Avro schema

Download Avro

```
set JAVA_HOME=%ProgramFiles%\Java\jdk1.7.0_79

cd D:\JAVA_PROJECTS\apache-kafka\src\main\java\schemas

java -jar D:\avro-tools-1.10.0.jar compile schema user_schema.avsc .
"%ProgramFiles%\Java\jdk1.8.0_25\bin\java" -jar D:\avro-tools-1.10.0.jar compile
schema user_schema.avsc .
```

5.2. Schema Registry

5.2.1. Schema registry with docker-compose

https://github.com/lensesio/fast-data-dev

```
<iframe width="560" height="315" src="https://www.youtube.com/watch?v=O8T7AUxhoKo"
frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-
in-picture" allowfullscreen></iframe>
```

```
https://github.com/ackintosh/kafka-connect-colormeshop/blob/master/docker-compose.yml

wget https://raw.githubusercontent.com/ackintosh/kafka-connect-
colormeshop/master/docker-compose.yml

docker-compose up kafka-cluster

http://192.168.56.10:3030/
```

5.2.1.1. Create topic in schema registry

```
https://youtu.be/O8T7AUxhoKo?t=359
docker run --rm -it --net=host landoop/fast-data-dev bash
```

```
kafka-topics --create --topic demo-kafka-connect --partitions 3 --replication-factor 1
--zookeeper 127.0.0.1:2181
```

5.2.2. Create file connector in schema registry

```
name=file-stream-demo-distributed
connector.class=org.apache.kafka.connect.file.FileStreamSourceConnector
tasks.max=1
file=demo-file.txt
topic=demo-kafka-connect
key.converter=org.apache.kafka.connect.json.JsonConverter
key.converter.schemas.enable=true
value.converter=org.apache.kafka.connect.json.JsonConverter
value.converter.schemas.enable=true
```

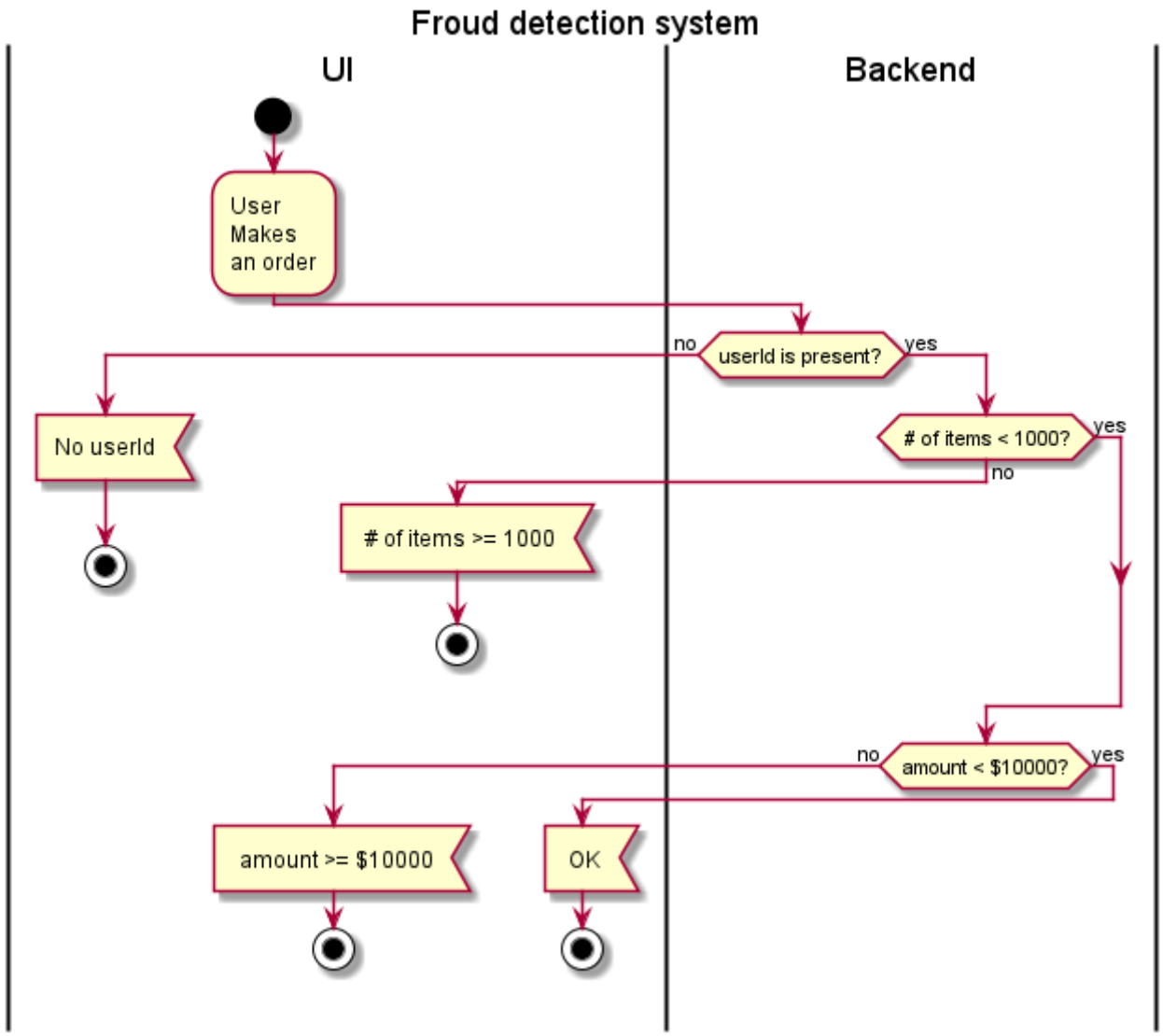
```
http://192.168.56.10:3030/kafka-topics-ui/#/cluster/fast-data-dev/topic/n/demo-kafka-
connect/

docker ps
docker exec -it <containerID> bash
touch demo-file.txt
echo "hi" >> demo-file.txt
echo "ho" >> demo-file.txt
echo "hu" >> demo-file.txt
```

```
sudo yum install git -y
sudo yum install java-11-openjdk-devel -y
sudo yum install maven -y
git clone https://github.com/confluentinc/schema-registry
cd schema-registry]
git checkout v5.2.0
mvn package
```

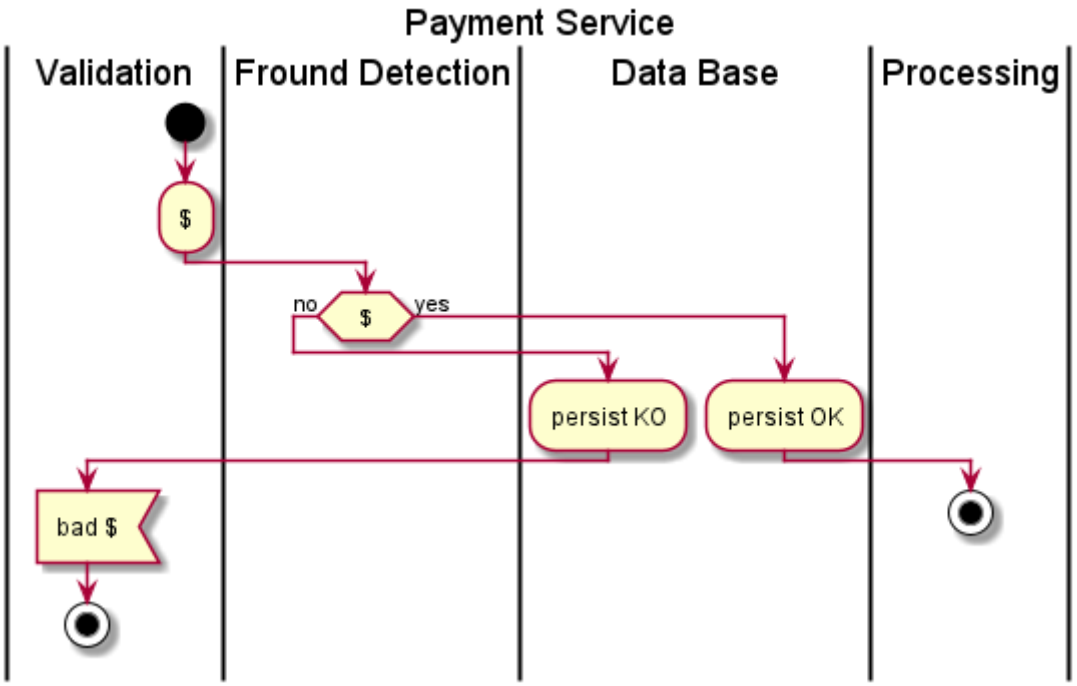
```
wget https://raw.githubusercontent.com/obsidiandynamics/kafdrop/master/docker-
compose/kafka-kafdrop/docker-compose.yaml
docker-compose up
http://192.168.56.10:9000
```

6. Streaming

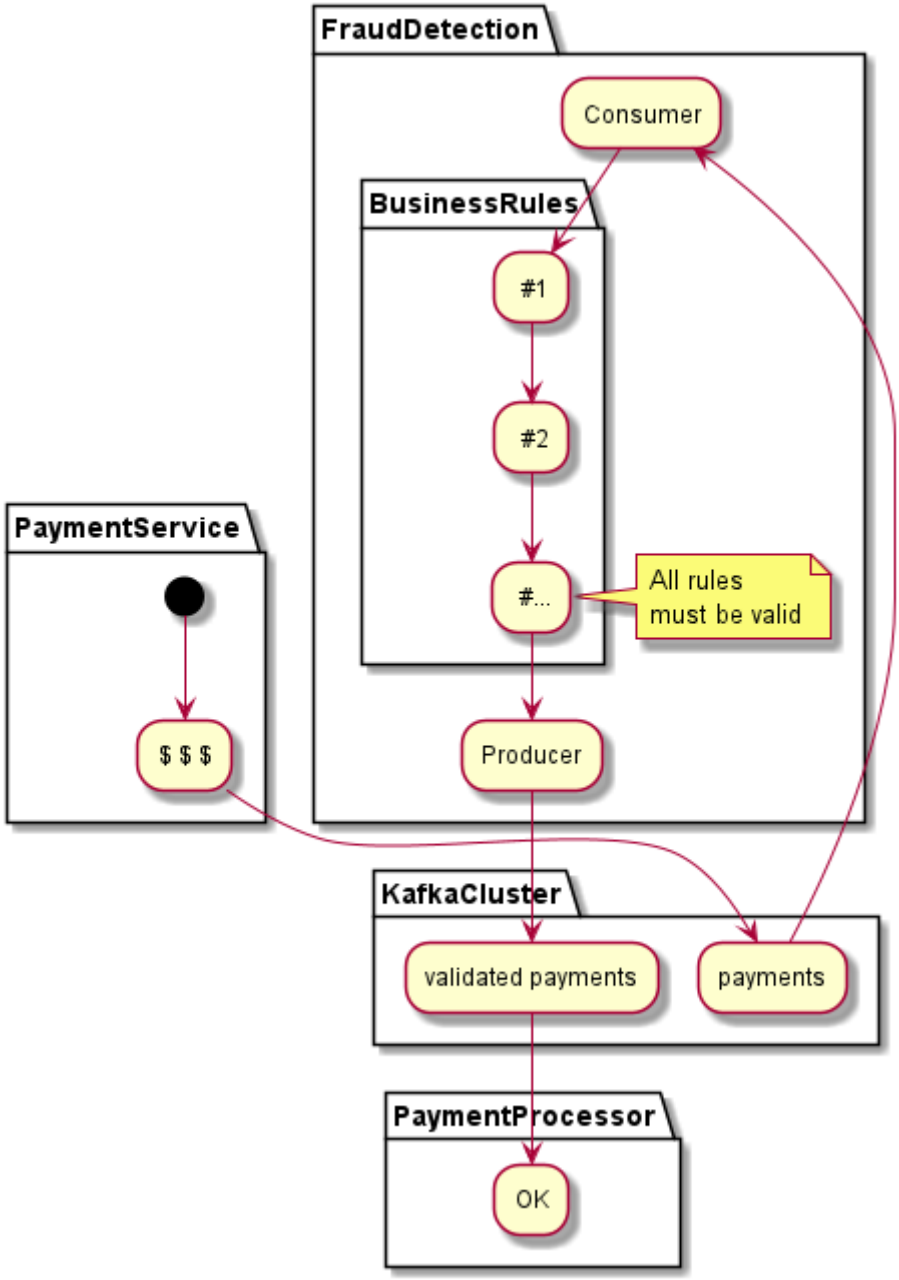
6.1. Froud detection system



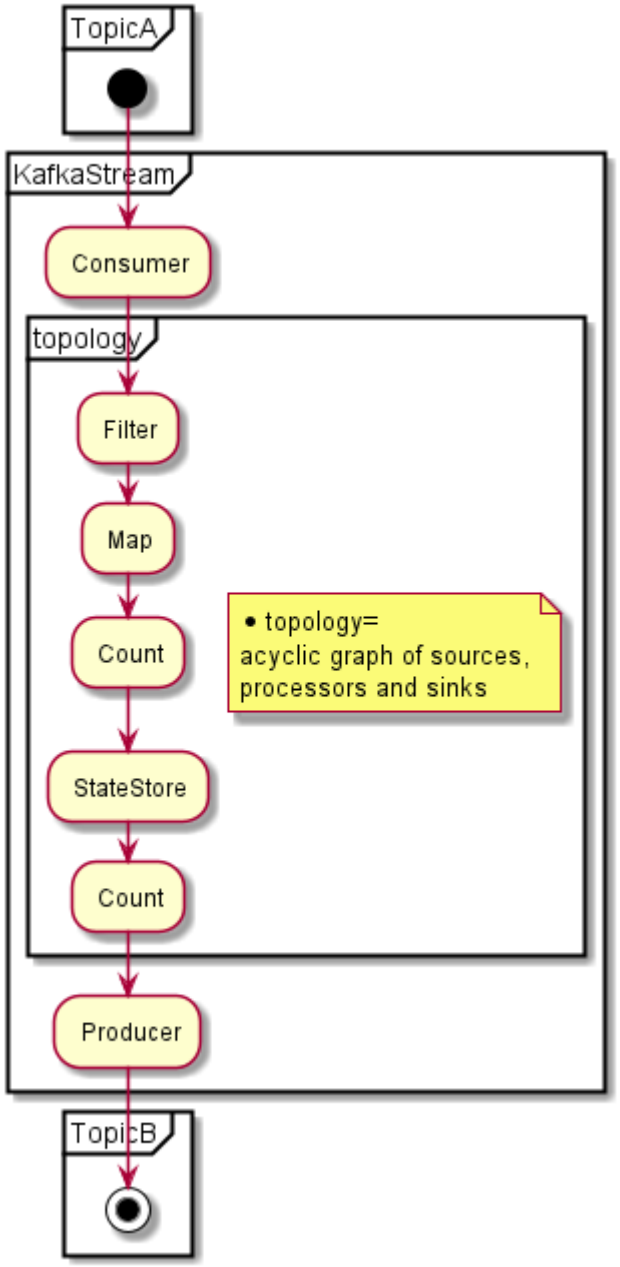
6.1.1. Traditional Design



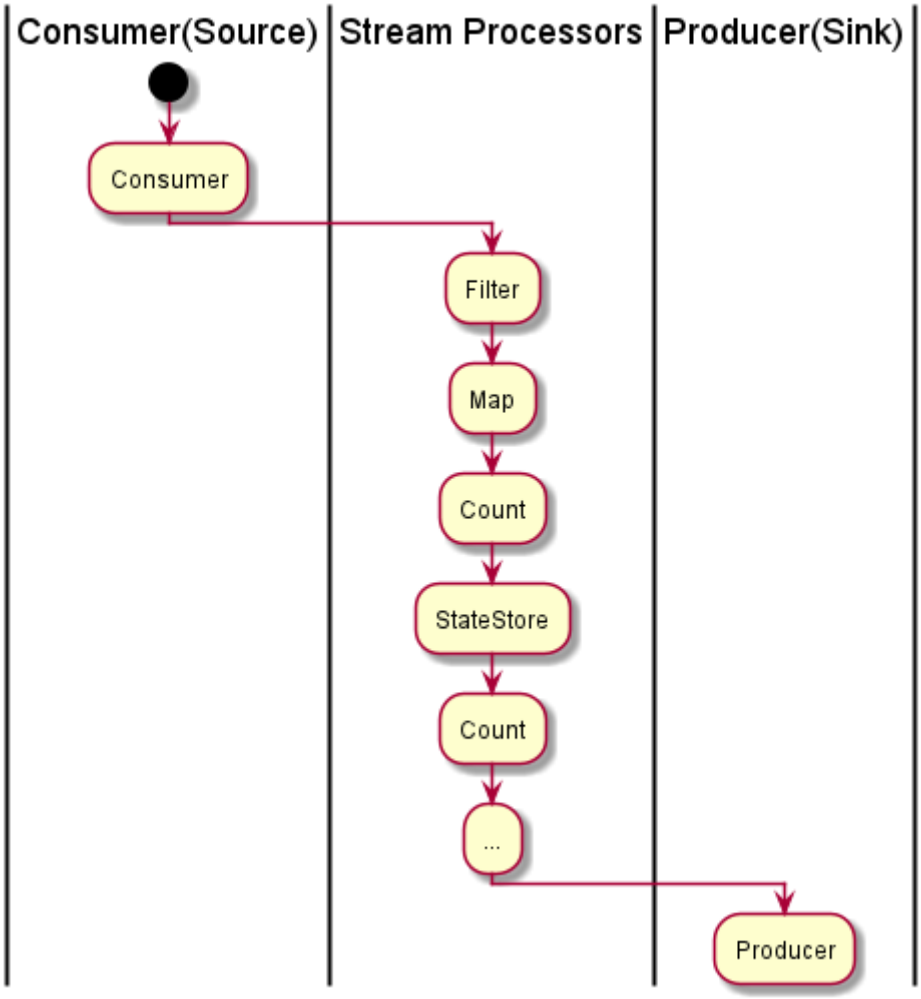
6.1.2. Streaming with Kafka



6.1.3. Kafka Streams



6.1.4. Stream Topology



6.1.5. Stateless Operations

Stateless Transformations @**kafka.apache.org**

- Branch
- Filter
- Inverse Filter
- Map
- FlatMap
- Foreach
- Peek
- GroupBy
- Merge

6.1.6. Stateful Operations

Stateful Transformations @**kafka.apache.org**

- Aggregation
- Count

- Joins
- Windowing
- Custom processors

7. KSQL

7.1. Windowing

- What is the average number of users visiting our website for hour or whtat is a total number of users which orders a product per day?

7.1.1. Tumbling

7.1.2. Hopping

7.1.3. KSQL Statements

7.1.3.1. Data definition language(DDL)

- CREATE STREAM
- CREATE TABLE
- DROP STREAM
- DROP TABLE
- CREATE STREAM AS SELECT
- CREATE TABLE AS SELECT

7.1.3.2. Data manipulation language(DML)

- SELECT
- INSERT
- CREATE STREAM AS SELECT
- CREATE TABLE AS SELECT

7.1.3.3. Alerting system

- "payments" → "warnings"
- Transactions > 5/10(mins window)

7.1.3.4. Confluent KSQL

```
# execute
/root/confluent-6.0.0/bin/ksql
```

```
ksql>
SHOW STREAMS;
SHOW TOPICS;
# create new topic
CREATE STREAM ksql_payments WITH ( KAFKA_TOPIC='payments', VALUE_FORMAT='AVRO' );
CREATE TABLE warnings AS SELECT userId, COUNT(*) FROM ksql_payments WINDOW HOOPING (
SIZE 10 MINUTES, ADVANCE BY 1 MINUTE ) GROUP BY userId HAVING COUNT(*) > 5;
PRINT 'payments';
PRINT 'WARNINGS';
```

7.1.4. Installation KSQL

```
git clone https://github.com/confluentinc/ksql.git
cd ksql
git checkout v.5.2.0
git checkout 5.5.x
mvn clean compile install -DskipTests

mvn archetype:generate -X \
    -DarchetypeGroupId=io.confluent.ksql \
    -DarchetypeArtifactId=ksql-udf-quickstart \
    -DarchetypeVersion=5.3.0

mvn package
vim config/ksql-server.properties
bootstrap.servers=localhost:9092
start zoopkeeper, broker(s), schema registry
bin/ksql-server-start config/ksql-server.properties
```

8. Kafdrop

```
kafka-topics --create --topic quickstart-events --bootstrap-server 192.168.56.10:9092
kafka-topics --create --topic quickstart-events --bootstrap-server localhost:9092
bin/kafka-topics.sh --create --topic quickstart-events --bootstrap-server
192.168.56.10:9092
```

http://192.168.56.10:9000/

```
docker run --rm -it --net=host confluentinc/cp-kafka:5.5.0 bash
kafka-topics --create --topic demo-kafka-connect --partitions 3 --replication-factor 1
--zookeeper 127.0.0.1:2181
kafka-topics --create --topic quickstart-events --partitions 3 --replication-factor 1
--zookeeper 127.0.0.1:2181
```

```
kafka-topics --create --topic quickstart-events --bootstrap-server sample-kafka:9092
kafka-console-producer --topic quickstart-events --bootstrap-server 127.0.0.1:2181
```

```
kafka-topics --bootstrap-server --zookeeper 127.0.0.1:2181 -create --partitions 3
--replication-factor 1 --topic streams-intro
```

```
kafka-topics --create --topic streams-intro --partitions 3 --replication-factor 1
--zookeeper 127.0.0.1:2181
```

```
kafka-console-producer --broker-list sample-kafka:9092 --topic streams-intro
--zookeeper 127.0.0.1:2181
```

```
kafka-console-producer --zookeeper 127.0.0.1:2181 --topic streams-intro --property
\"parse.key=true\" --property \"key.separator=:\"
```

```
kafka-console-producer --zookeeper 127.0.0.1:2181 --topic streams-intro --property
"parse.key=true" --property "key.separator=:"
```

```
kafka-topics --create --topic quickstart-events --bootstrap-server 192.168.56.10:9092
```