Building a sophisticated stream topology

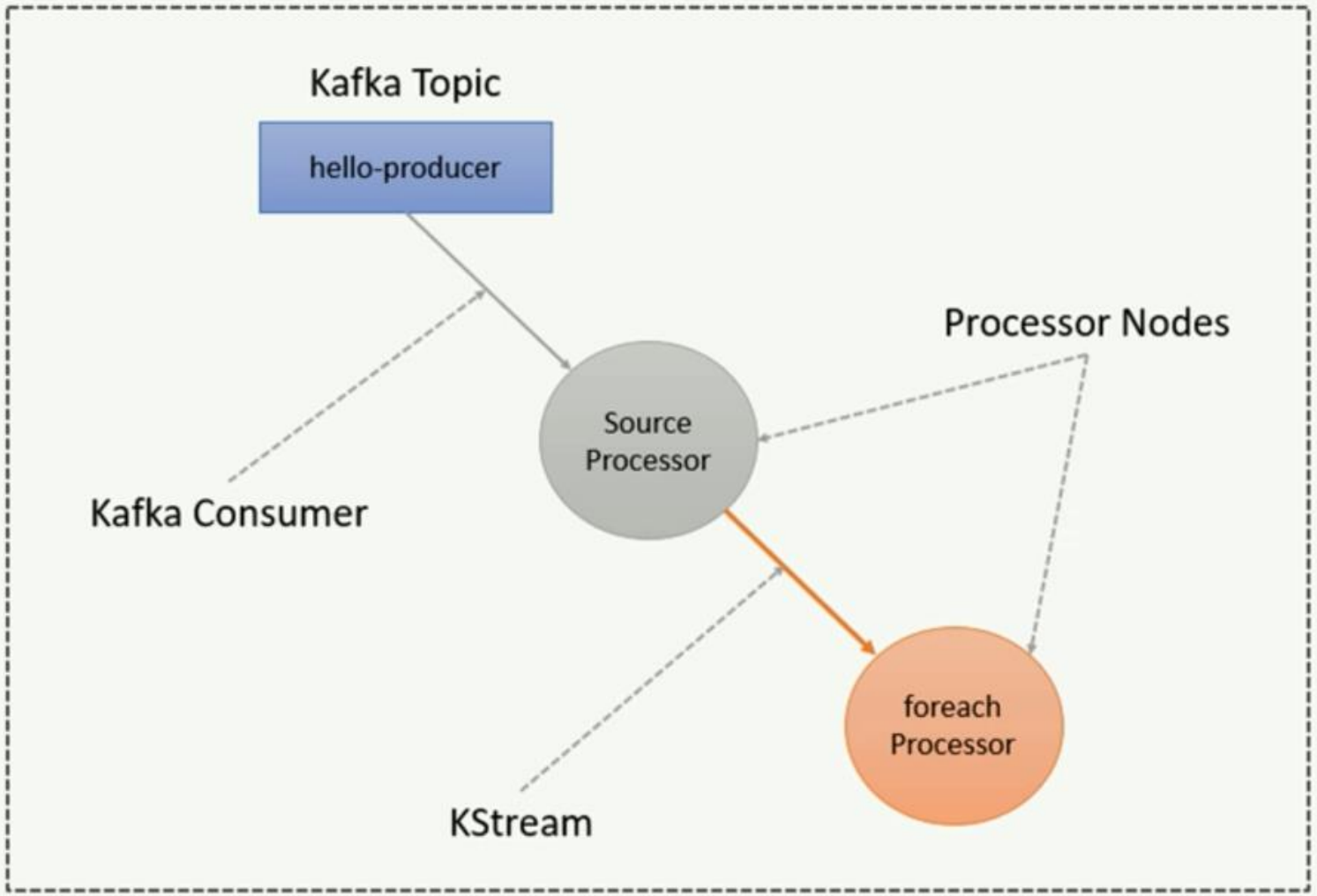
Creating Kafka Streams Application

- Create Configurations.
- Create Streams Topology.
- Start the Stream.
- Add shutdown hook.

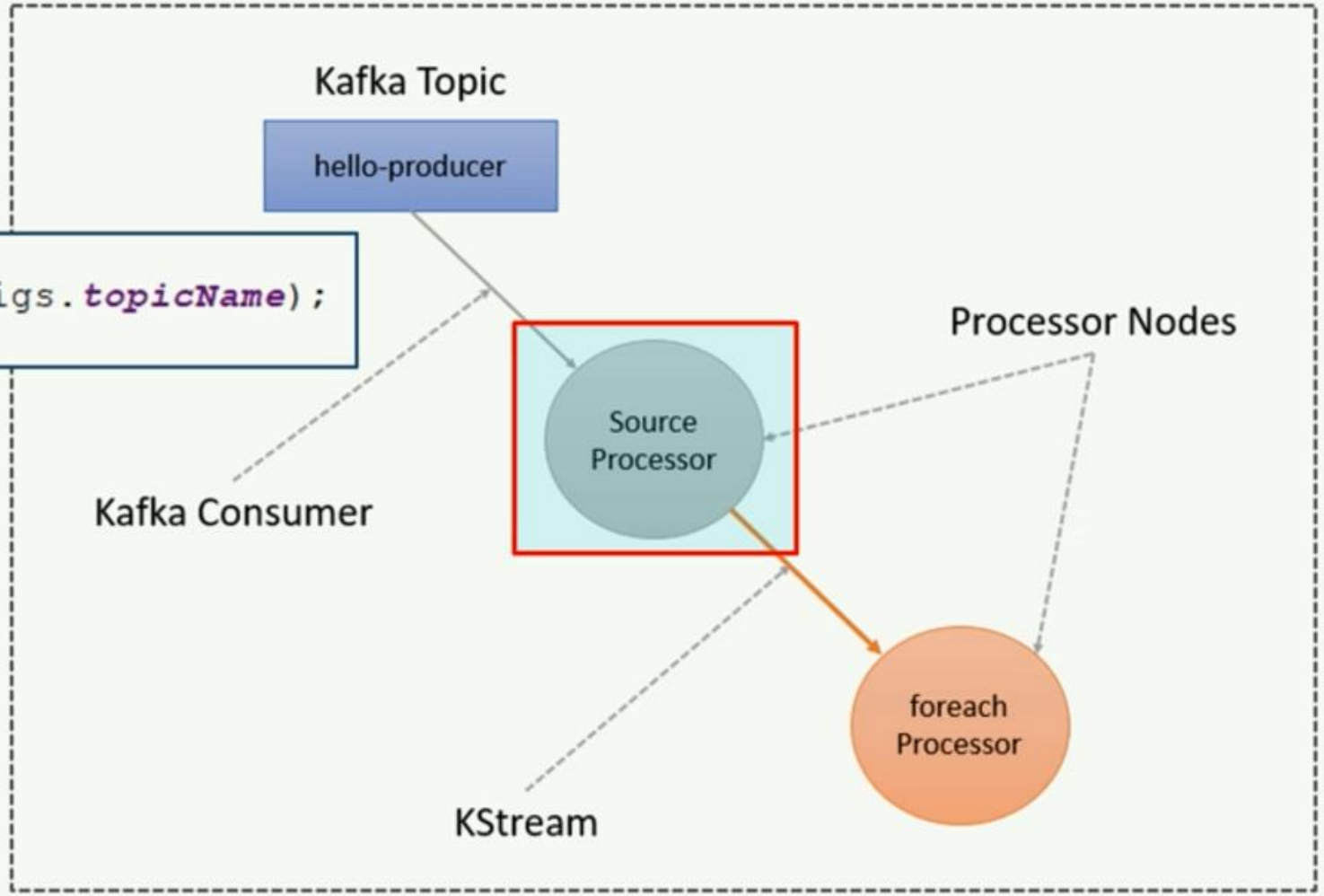
What is Streams Topology?

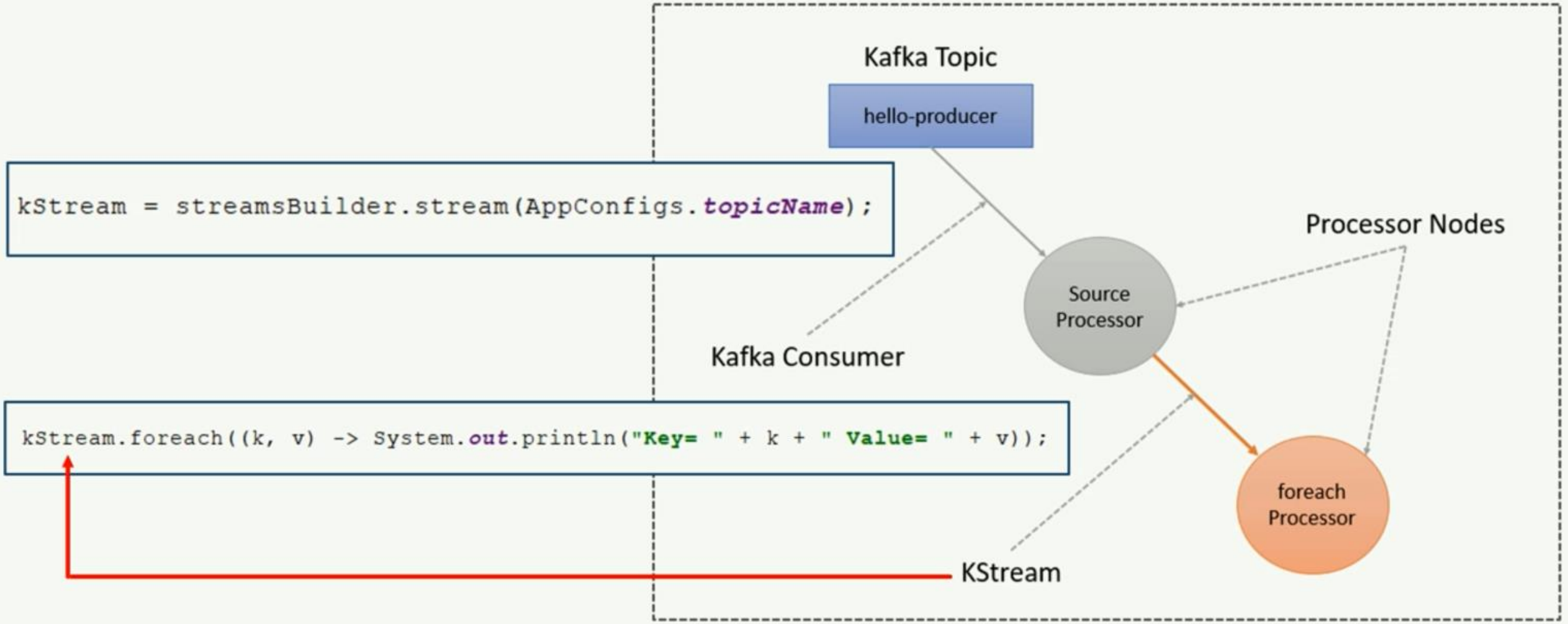
Step-by-step computational logic of a stream processing application.

Topology DAG



```
kStream = streamsBuilder.stream(AppConfigs.topicName);
```




Kafka Topic
hello-producer
kStream = streamsBuilder.stream(AppConfigs.topicName);
Processor Nodes
Source Processor
Kafka Consumer
kStream.foreach((k, v) -> System.out.println("Key= " + k + " Value= " + v));
foreach Processor
KStream

KStream

Abstraction of a stream of Kafka message records.

filter()
map()
flatmap()
foreach()
to()

KStream

Abstraction of a stream of Kafka message records.

The void method performs some actions or operations but does not produce a result that can be used or accessed by the caller.

Problem Statement

- Use the POS Simulator Application to generate a continuous stream of Invoices.
- Create a stream processing application for the following

Business Requirement

XYZ is a Home Furniture and Kitchen utensils retailer. They have twenty retail stores spread all over the country. XYZ management decided to transform themselves into a real-time data-driven organization. As a first step towards that goal, they started sending their invoices to a Kafka cluster in real-time. The POS machines in all their stores are now sending invoices to a Kafka topic – POS. As a next step, they want to create the following automated services.

1. Shipment Service
2. Loyalty Management Service
3. Trend Analytics

While other teams are working on the implementation details of these three services, you are asked to create a Kafka Streams application that does following.

1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipment service queue.
2. Select Invoices where CustomerType = "PRIME" and create a notification event for the Loyalty Management Service. The format for the new notification event is given here.
3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.

Problem Statement

- Use the POS Simulator Application to generate a continuous stream of Invoices.
- Create a stream processing application for the following

Business Requirement

XYZ is a Home Furniture and Kitchen utensils retailer. They have twenty retail
management decided to transform themselves into a real-time data-driven organizati
started sending their invoices to a Kafka cluster in real-time. The POS machines in all
Kafka topic – POS. As a next step, they want to create the following automated service

1. Shipment Service
2. Loyalty Management Service
3. Trend Analytics

While other teams are working on the implementation details of these three services, you are asked to create a Kafka Streams application that does following.

1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipment service queue.
2. Select Invoices where CustomerType = "PRIME" and create a notification event for the Loyalty Management Service. The format for the new notification event is given here.
3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.

```
{
  "InvoiceNumber": "xx",
  "CustomerCardNo": "xx",
  "TotalAmount": "xx",
  "EarnedLoyaltyPoints": "xx"
}
```

Problem Statement

- Use the POS Simulator Application to generate a continuous stream of Invoic
- Create a stream processing application for the following

Business Requirement

XYZ is a Home Furniture and Kitchen utensils retailer. They have twenty retail sto management decided to transform themselves into a real-time data-driven organization started sending their invoices to a Kafka cluster in real-time. The POS machines in all the Kafka topic – POS. As a next step, they want to create the following automated services.

1. Shipment Service
2. Loyalty Management Service
3. Trend Analytics

While other teams are working on the implementation details of these three services, y application that does following.

1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipr
2. Select Invoices where CustomerType = "PRIME" and create a notification event for format for the new notification event is given here.
3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.


```
{
  "InvoiceNumber": "xx",
  "CreatedTime": "xx",
  "StoreID": "xx",
  "PosID": "xx",
  "CustomerType": "xx",
  "PaymentMethod": "xx",
  "DeliveryType": "xx",
  "City": "xx",
  "State": "xx",
  "PinCode": "xx",
  "ItemCode": "xx",
  "ItemDescription": "xx",
  "ItemPrice": "xx",
  "ItemQty": "xx",
  "TotalValue": "xx"
}
```

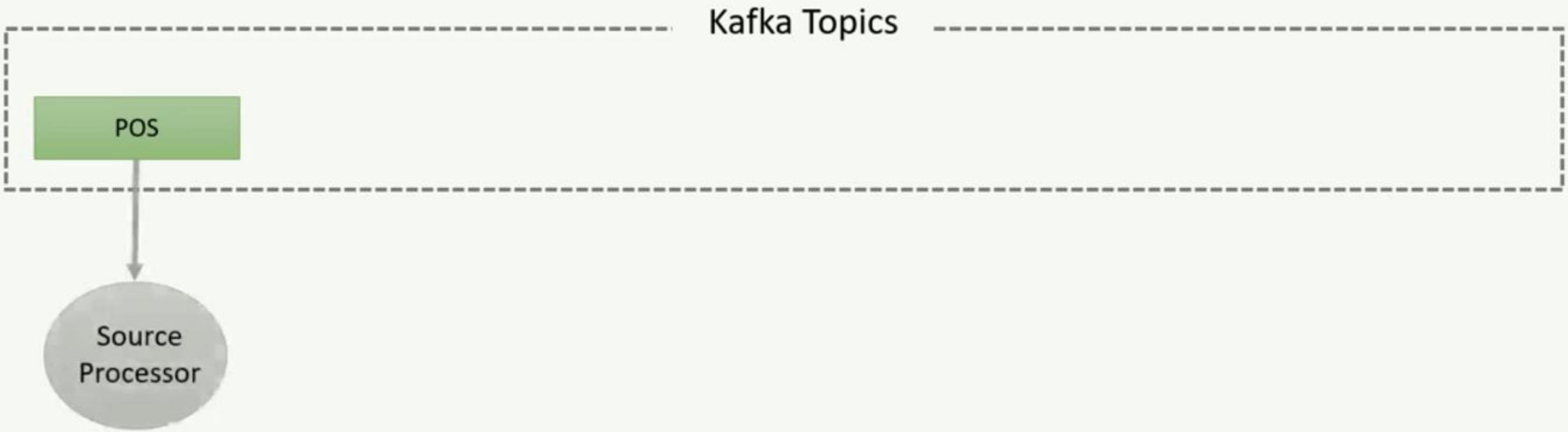
Requirement Extract

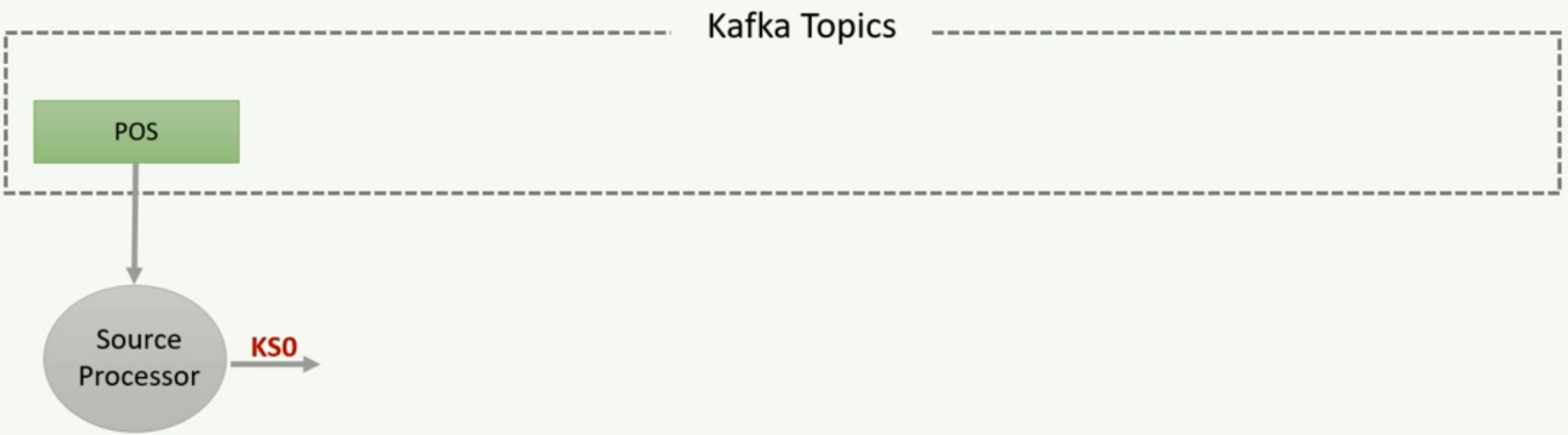
1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipment service queue.

2. Select Invoices where CustomerType = "PRIME" and create a notification event for the Loyalty Management Service. The format for the new notification event is given here.

3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.


Kafka Topics
POS


Kafka Topics
POS
Source
Processor

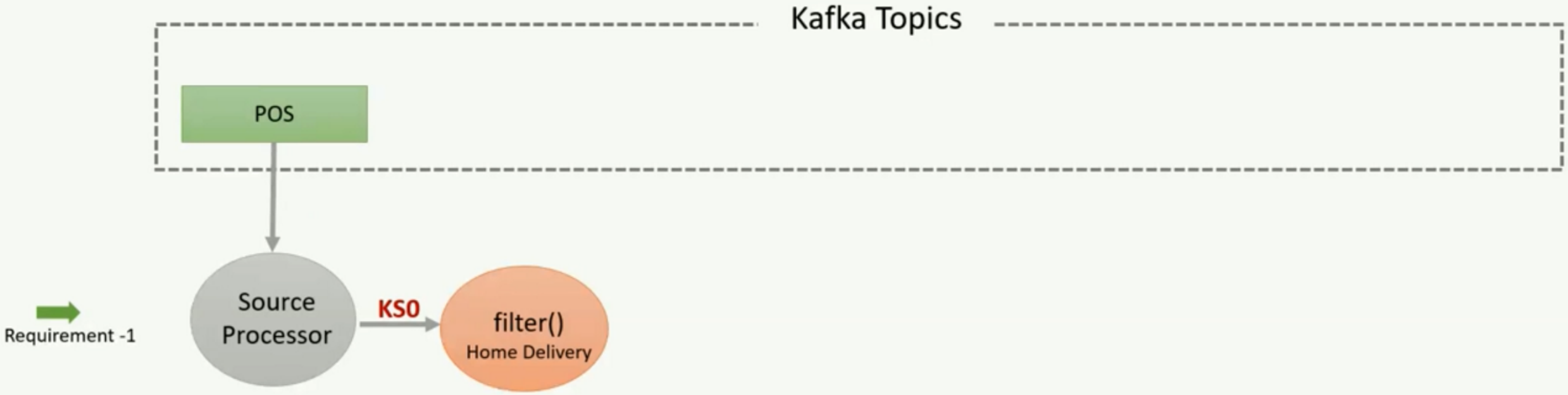

Kafka Topics
POS
Source
Processor
KS0

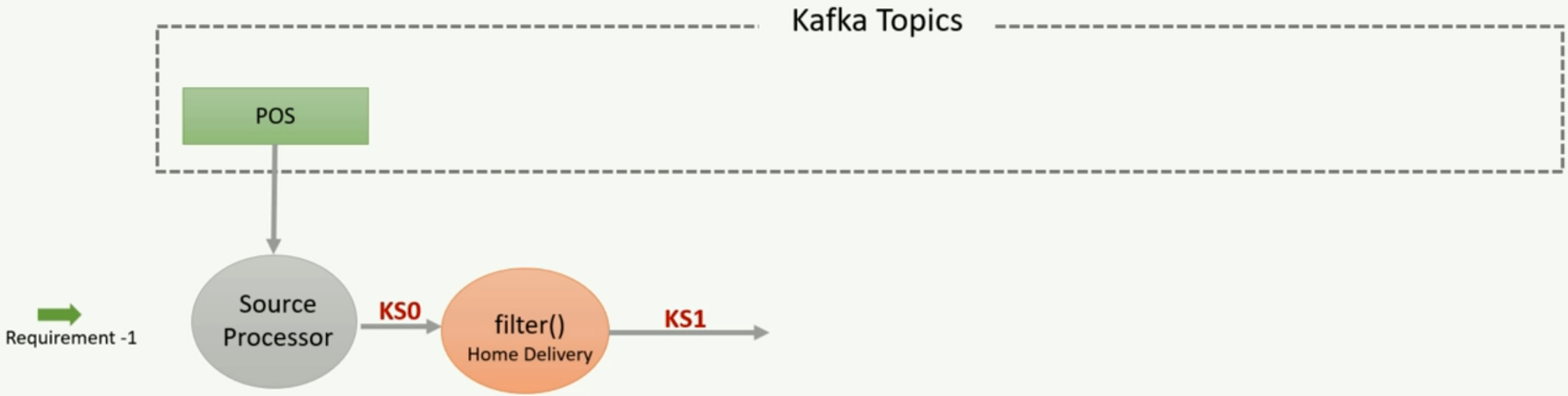
Requirement Extract

1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipment service queue.

2. Select Invoices where CustomerType = "PRIME" and create a notification event for the Loyalty Management Service. The format for the new notification event is given here.

3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.


Kafka Topics
POS
Source Processor
KS0
filter()
Home Delivery
Requirement -1

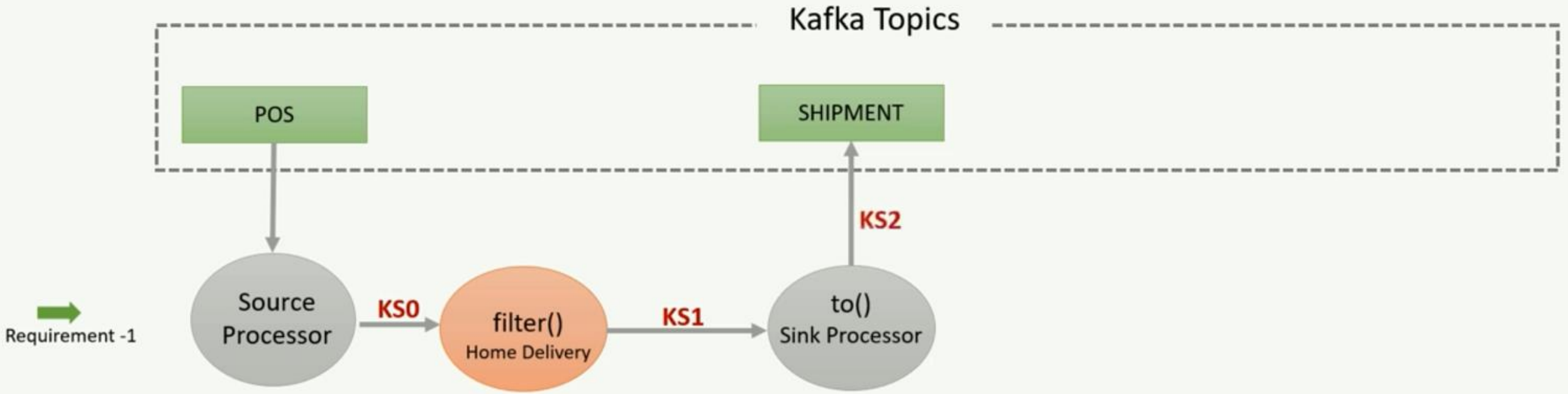

Kafka Topics
POS
Source Processor
KS0
filter()
Home Delivery
KS1
Requirement -1

Requirement Extract

1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipment service queue.

2. Select Invoices where CustomerType = "PRIME" and create a notification event for the Loyalty Management Service. The format for the new notification event is given here.

3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.

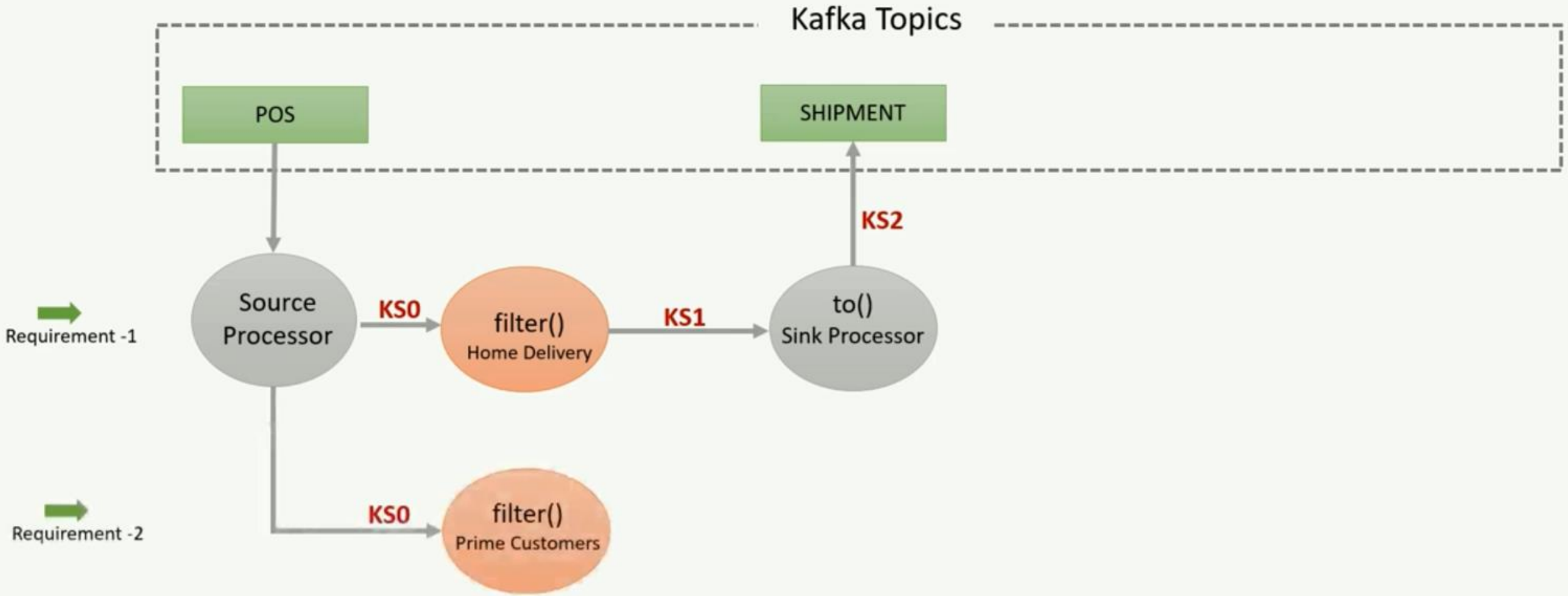

Kafka Topics
POS
SHIPMENT
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
Requirement -1

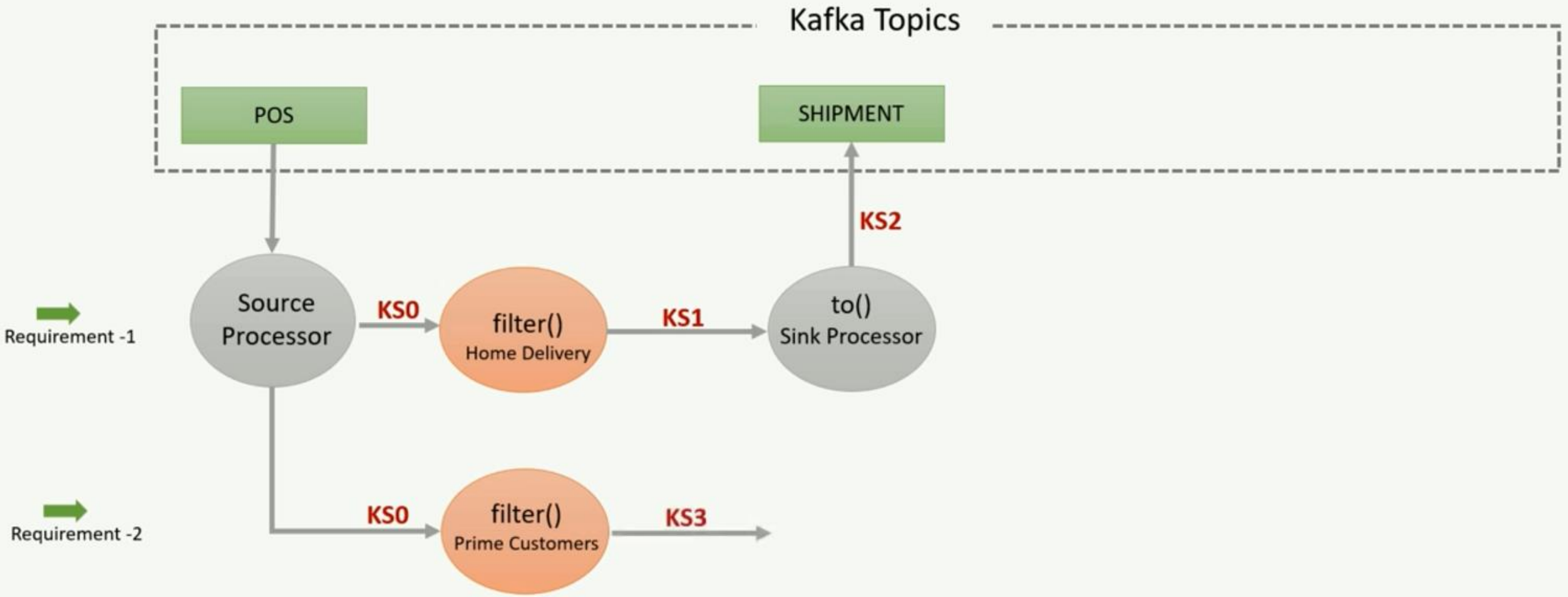
Requirement Extract

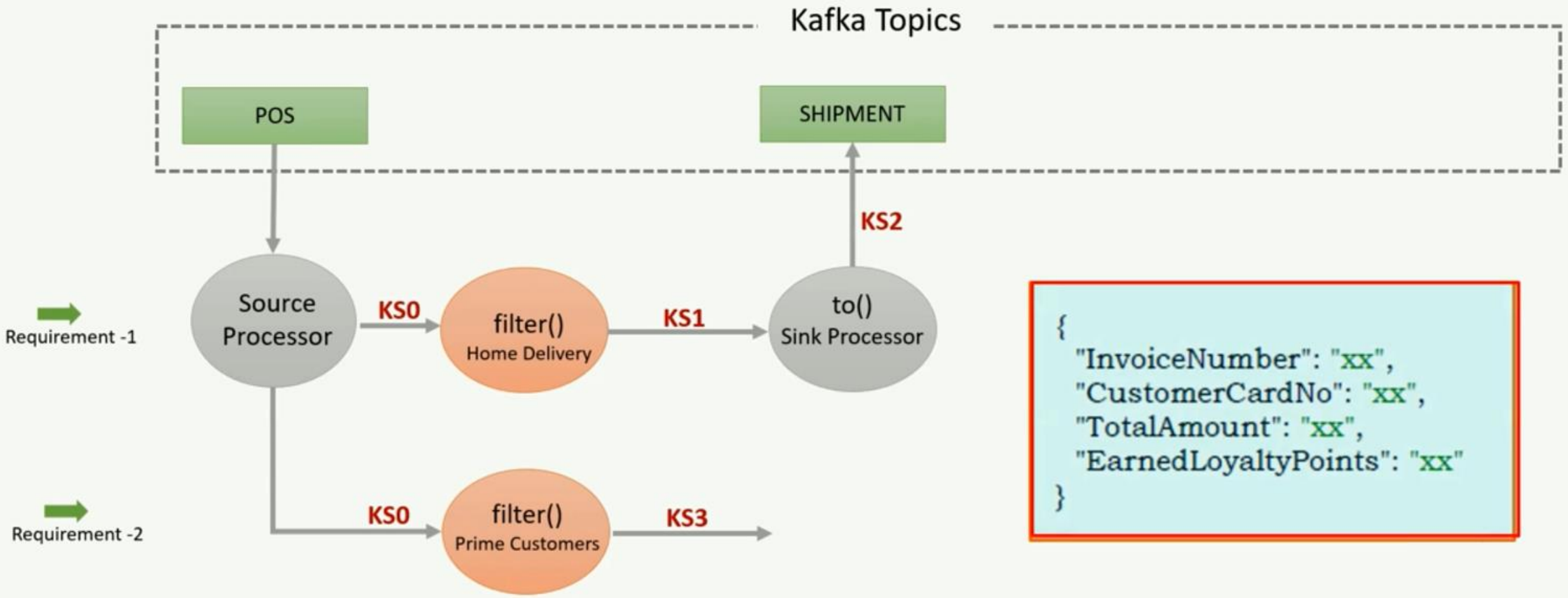
1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipment service queue.

2. Select Invoices where CustomerType = "PRIME" and create a notification event for the Loyalty Management Service. The format for the new notification event is given here.

3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.

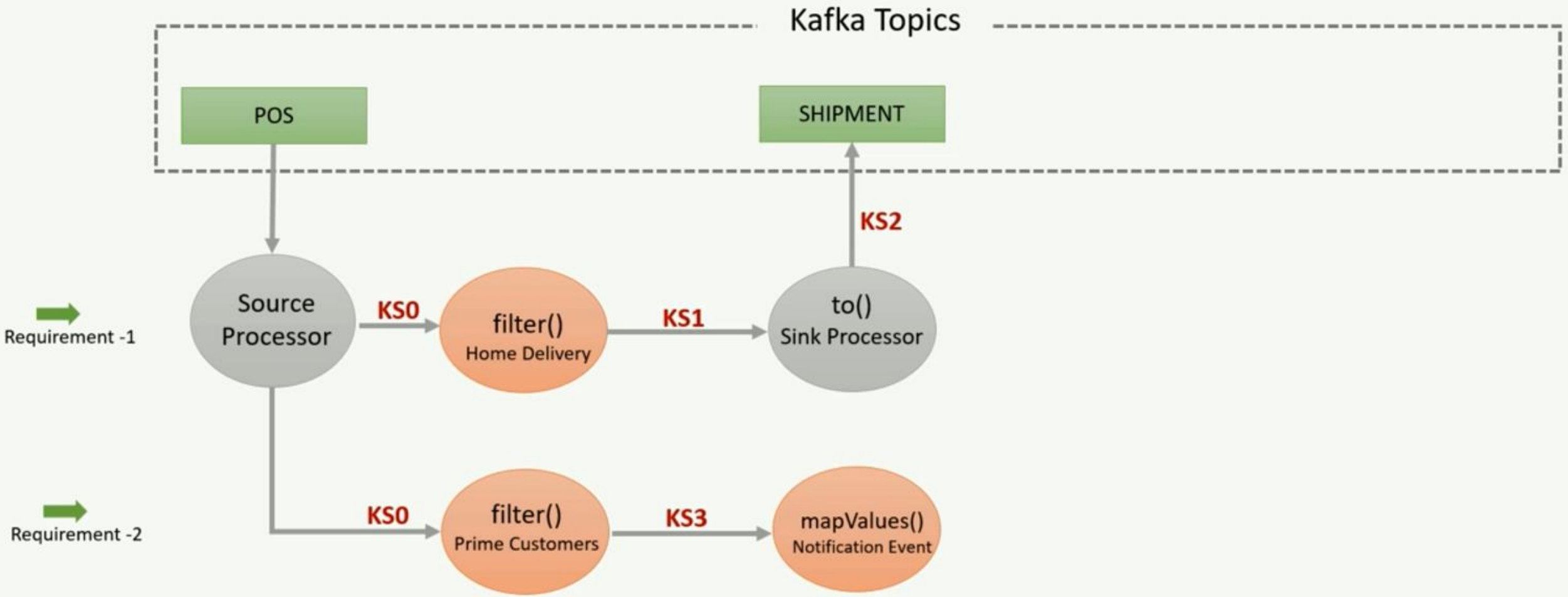

Kafka Topics
POS
SHIPMENT
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
Requirement -1
KS0
filter()
Prime Customers
Requirement -2

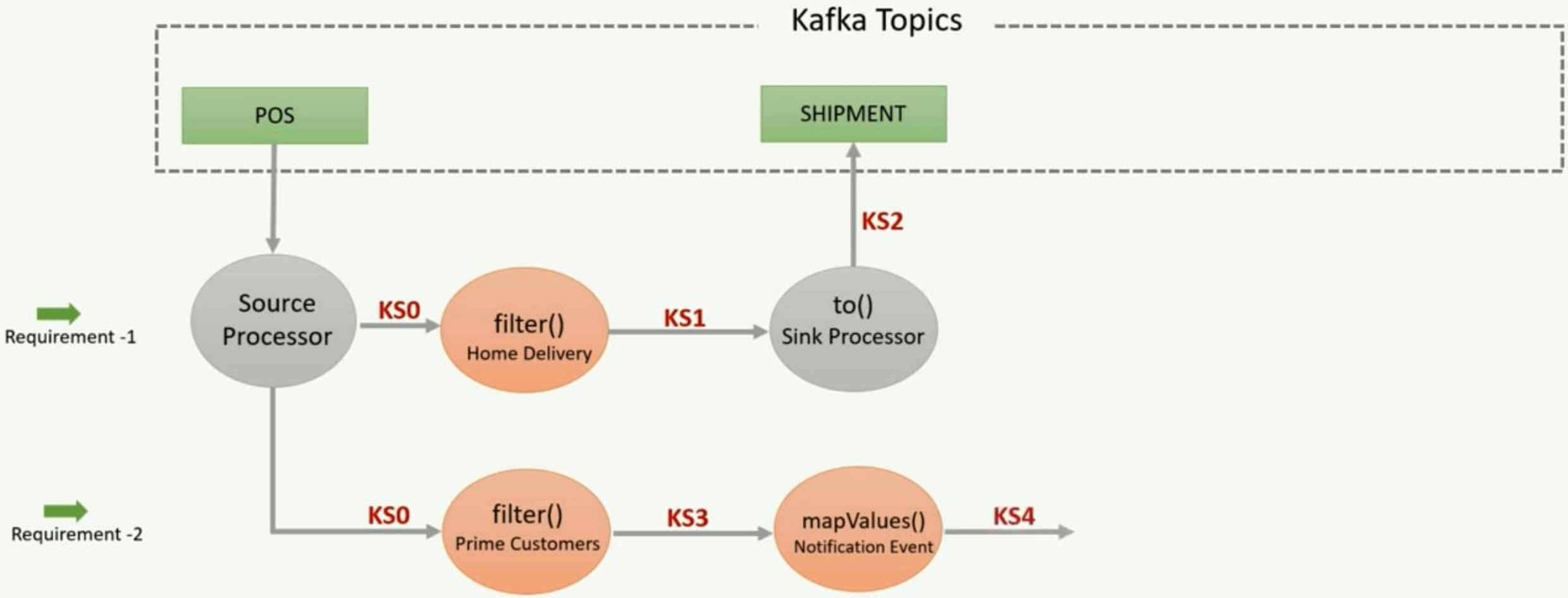

Kafka Topics
POS
SHIPMENT
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
Requirement -1
KS0
filter()
Prime Customers
KS3
Requirement -2


Kafka Topics
POS
SHIPMENT
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
Requirement -1
Requirement -2
KS0
filter()
Prime Customers
KS3
{
"InvoiceNumber": "xx",
"CustomerCardNo": "xx",
"TotalAmount": "xx",
"EarnedLoyaltyPoints": "xx"
}

```
{
  "type": "object",
  "javaType": "ir.mfozouni.types.Notification",
  "properties": {
    "InvoiceNumber": {
      "type": "string"
    },
    "CustomerCardNo": {
      "type": "string"
    },
    "TotalAmount": {
      "type": "number"
    },
    "EarnedLoyaltyPoints": {
      "type": "number"
    }
  }
```

```
{
  "InvoiceNumber": "xx",
  "CustomerCardNo": "xx",
  "TotalAmount": "xx",
  "EarnedLoyaltyPoints": "xx"
}
```


Kafka Topics
POS
SHIPMENT
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
Requirement -1
KS0
filter()
Prime Customers
KS3
mapValues()
Notification Event
Requirement -2

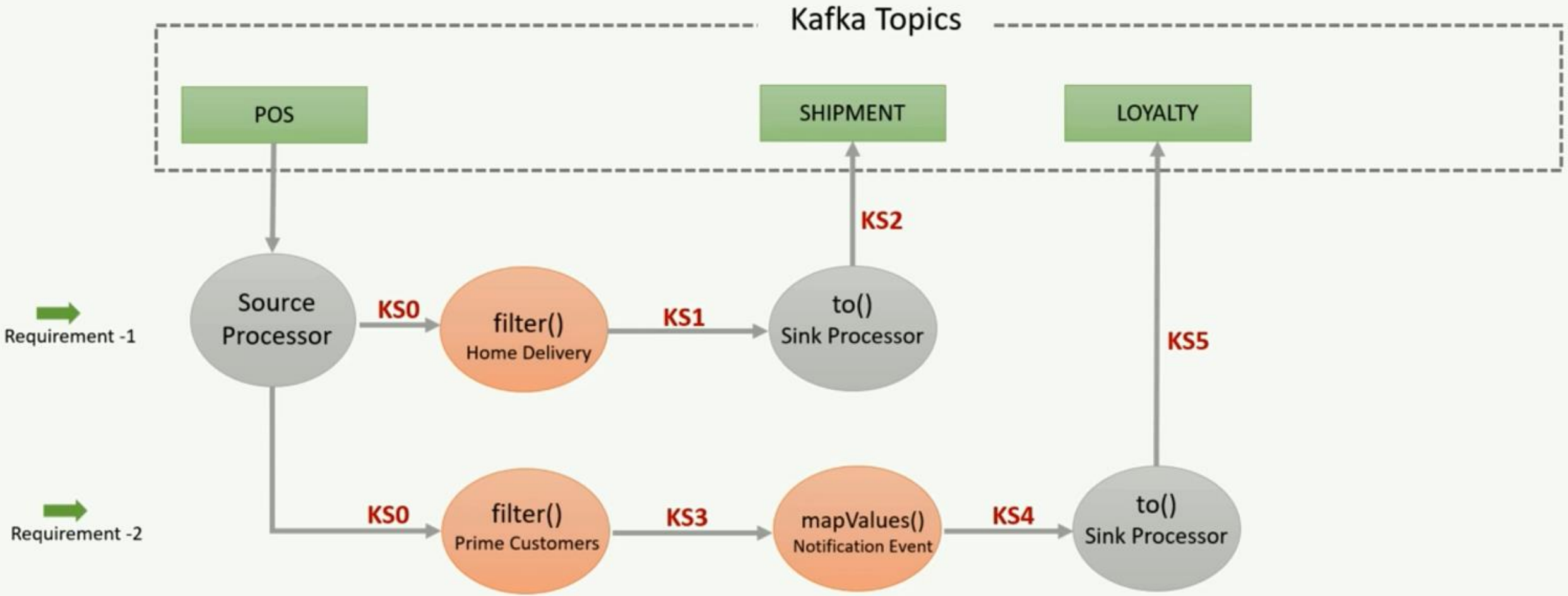

Kafka Topics
POS
SHIPMENT
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
Requirement -1
KS0
filter()
Prime Customers
KS3
mapValues()
Notification Event
KS4
Requirement -2

Requirement Extract

1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipment service queue.

2. Select Invoices where CustomerType = "PRIME" and create a notification event for the Loyalty Management Service. The format for the new notification event is given here.

3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.

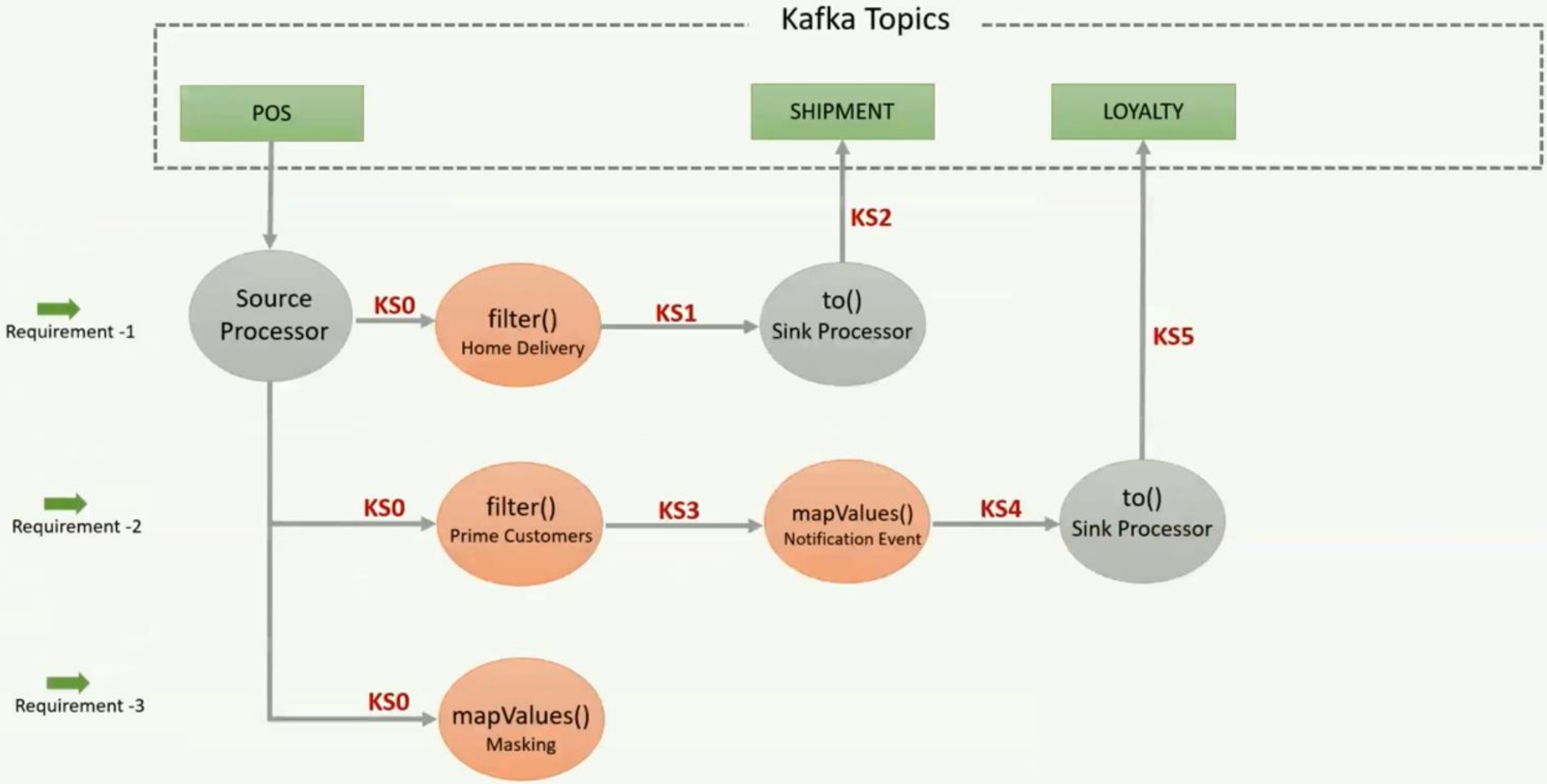

Kafka Topics
POS
SHIPMENT
LOYALTY
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
KS5
Requirement -1
Requirement -2
KS0
filter()
Prime Customers
KS3
mapValues()
Notification Event
KS4
to()
Sink Processor

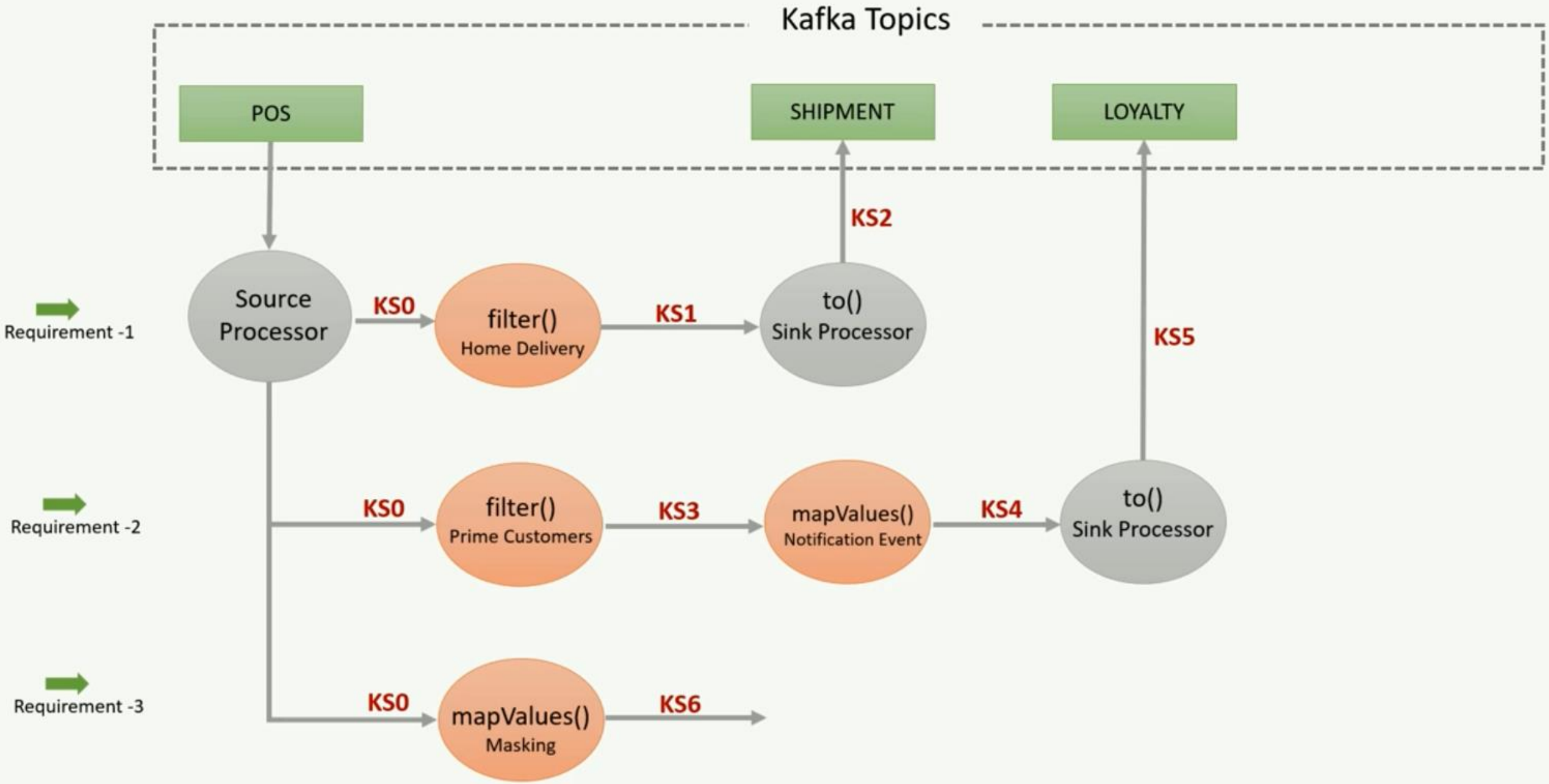
Requirement Extract

1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipment service queue.

2. Select Invoices where CustomerType = "PRIME" and create a notification event for the Loyalty Management Service. The format for the new notification event is given here.

3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.


Kafka Topics
POS
SHIPMENT
LOYALTY
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
KS5
Requirement -1
Requirement -2
KS0
filter()
Prime Customers
KS3
mapValues()
Notification Event
KS4
to()
Sink Processor
Requirement -3
KS0
mapValues()
Masking


Kafka Topics
POS
SHIPMENT
LOYALTY
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
KS5
Requirement -1
Requirement -2
KS0
filter()
Prime Customers
KS3
mapValues()
Notification Event
KS4
to()
Sink Processor
Requirement -3
KS0
mapValues()
Masking
KS6

Requirement Extract

1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipment service queue.

2. Select Invoices where CustomerType = "PRIME" and create a notification event for the Loyalty Management Service. The format for the new notification event is given here.

3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.


```
{
  "InvoiceNumber": "xx",
  "CreatedTime": "xx",
  "StoreID": "xx",
  "PosID": "xx",
  "CustomerType": "xx",
  "PaymentMethod": "xx",
  "DeliveryType": "xx",
  "City": "xx",
  "State": "xx",
  "PinCode": "xx",
  "ItemCode": "xx",
  "ItemDescription": "xx",
  "ItemPrice": "xx",
  "ItemQty": "xx",
  "TotalValue": "xx"
}
```

Sample Invoice

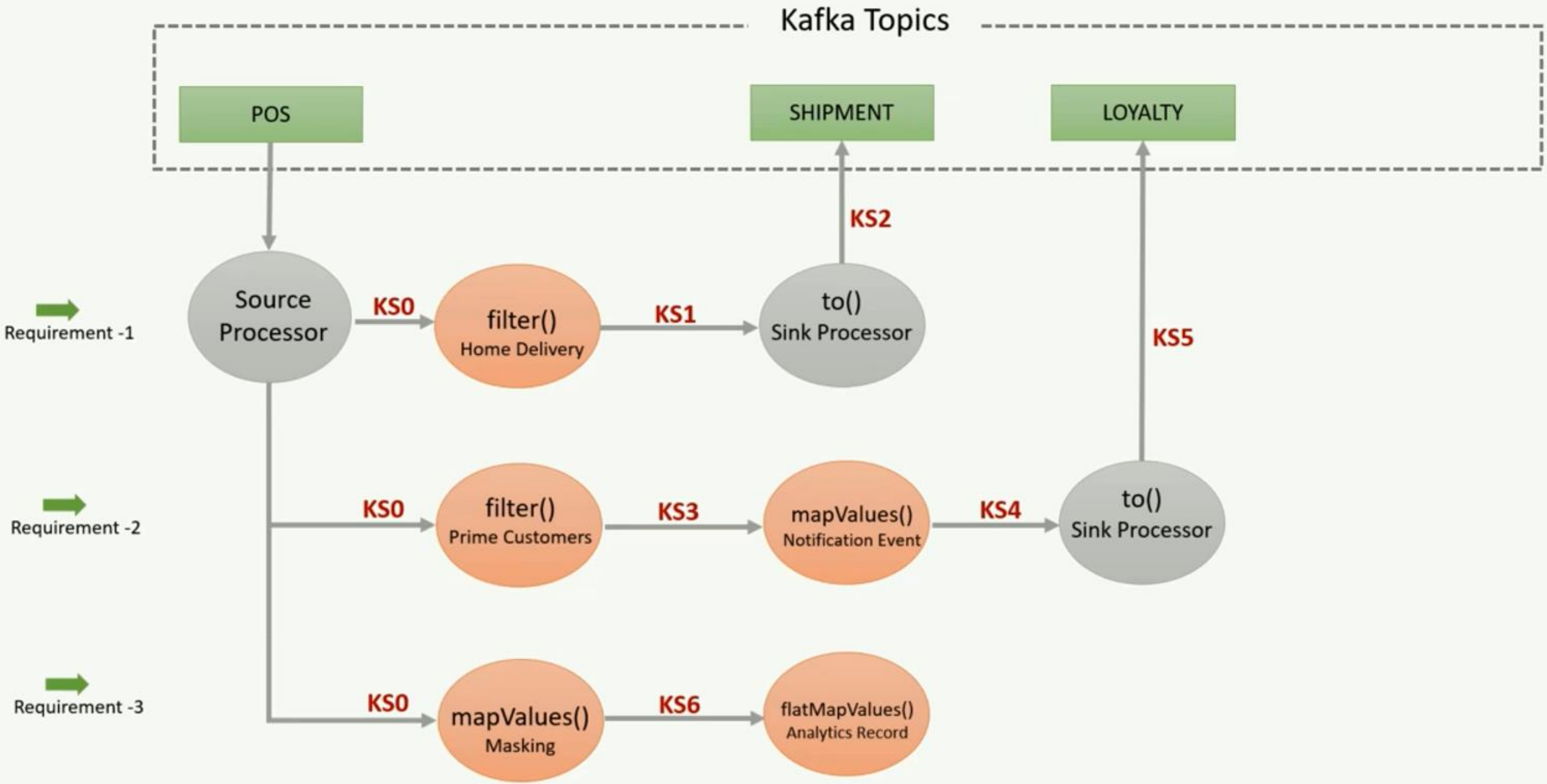

```
{  "InvoiceNumber":"2798495",
   "CreatedTime":1552393659955,
   "StoreID":"STR7188",
   "PosID":"POS825",
   "CashierID":"OAS329",
   "CustomerType":"PRIME",
   "CustomerCardNo":"7051101351",
   "TotalAmount":3529.0,
   "NumberOfItems":2,
   "PaymentMethod":"CASH",
   "TaxableAmount":3529.0,
   "CGST":88.22500000000001,
   "SGST":88.22500000000001,
   "CESS":4.41125,
   "DeliveryType":"HOME-DELIVERY",
   "DeliveryAddress":{
      "AddressLine":"House No 727, 9696 Ullamcorper, Road",
      "City":"Dabgram",
      "State":"West Bengal",
      "PinCode":"953658",
      "ContactNumber":"4166559042"
   },
   "InvoiceLineItems":[
      {  "ItemCode":"213",
         "ItemDescription":"Infant bed",
         "ItemPrice":1755.0,
         "ItemQty":1,
         "TotalValue":1755.0
      },
      {  "ItemCode":"628",
         "ItemDescription":"Window Scarf",
         "ItemPrice":1774.0,
         "ItemQty":1,
         "TotalValue":1774.0
      }
   ]
}
```

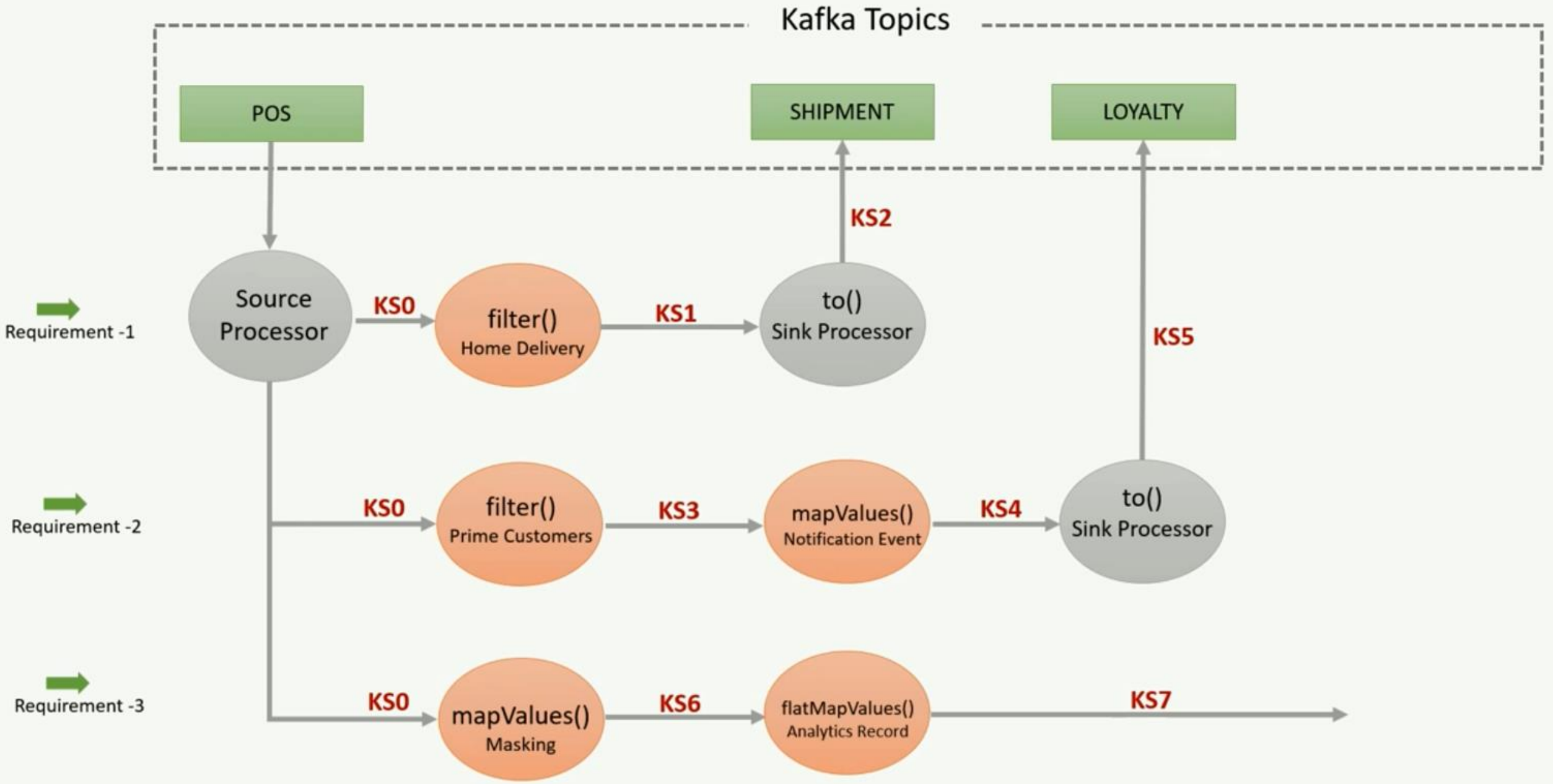
Sample Invoice


```
{  "InvoiceNumber":"2798495",
   "CreatedTime":1552393659955,
   "StoreID":"STR7188",
   "PosID":"POS825",
   "CashierID":"OAS329",
   "CustomerType":"PRIME",
   "CustomerCardNo":"7051101351",
   "TotalAmount":3529.0,
   "NumberOfItems":2,
   "PaymentMethod":"CASH",
   "TaxableAmount":3529.0,
   "CGST":88.22500000000001,
   "SGST":88.22500000000001,
   "CESS":4.41125,
   "DeliveryType":"HOME-DELIVERY",
   "DeliveryAddress":{
      "AddressLine":"House No 727, 9696 Ullamcorper, Road",
      "City":"Dabgram",
      "State":"West Bengal",
      "PinCode":"953658",
      "ContactNumber":"4166559042"
   },
   "InvoiceLineItems":[
      {  "ItemCode":"213",
         "ItemDescription":"Infant bed",
         "ItemPrice":1755.0,
         "ItemQty":1,
         "TotalValue":1755.0
      },
      {  "ItemCode":"628",
         "ItemDescription":"Window Scarf",
         "ItemPrice":1774.0,
         "ItemQty":1,
         "TotalValue":1774.0
      }
   ]
}
```



```
{
  "InvoiceNumber": "xx",
  "CreatedTime": "xx",
  "StoreID": "xx",
  "PosID": "xx",
  "CustomerType": "xx",
  "PaymentMethod": "xx",
  "DeliveryType": "xx",
  "City": "xx",
  "State": "xx",
  "PinCode": "xx",
  "ItemCode": "xx",
  "ItemDescription": "xx",
  "ItemPrice": "xx",
  "ItemQty": "xx",
  "TotalValue": "xx"
}
```


Kafka Topics
POS
SHIPMENT
LOYALTY
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
KS5
Requirement -1
Requirement -2
KS0
filter()
Prime Customers
KS3
mapValues()
Notification Event
KS4
to()
Sink Processor
Requirement -3
KS0
mapValues()
Masking
KS6
flatMapValues()
Analytics Record

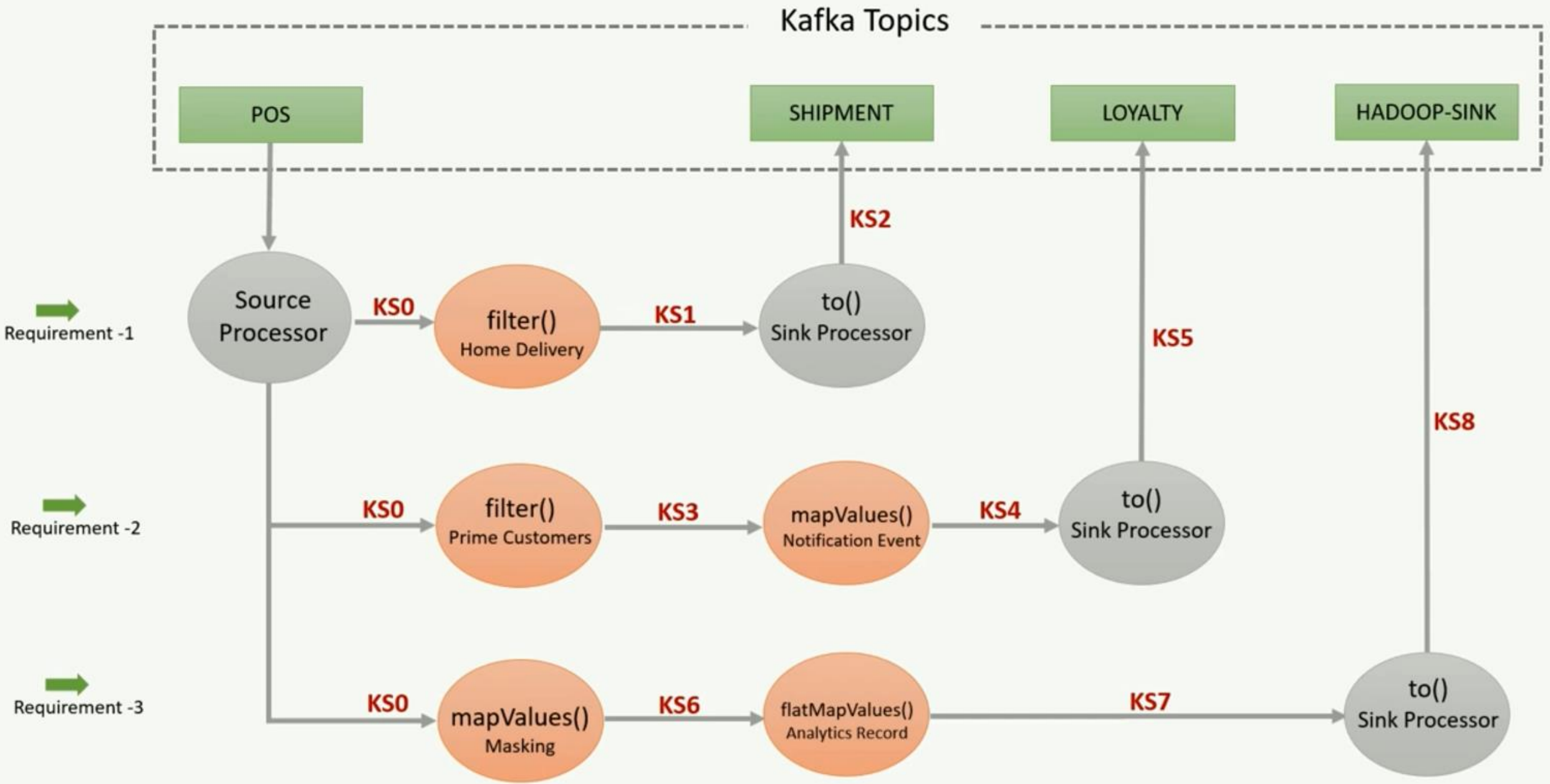

Kafka Topics
POS
SHIPMENT
LOYALTY
KS2
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
KS5
Requirement -1
Requirement -2
KS0
filter()
Prime Customers
KS3
mapValues()
Notification Event
KS4
to()
Sink Processor
Requirement -3
KS0
mapValues()
Masking
KS6
flatMapValues()
Analytics Record
KS7

Requirement Extract

1. Select Invoices where DeliveryType = "HOME-DELIVERY" and push them to the shipment service queue.

2. Select Invoices where CustomerType = "PRIME" and create a notification event for the Loyalty Management Service. The format for the new notification event is given here.

3. Select all Invoices, mask the personal information, and create records for Trend Analytics. When the records are ready, persist them to Hadoop storage for batch analytics. The format for the new Hadoop record is also given.


Kafka Topics
POS
SHIPMENT
LOYALTY
HADOOP-SINK
Source Processor
KS0
filter()
Home Delivery
KS1
to()
Sink Processor
KS2
KS5
KS8
Requirement -1
Requirement -2
Requirement -3
KS0
filter()
Prime Customers
KS3
mapValues()
Notification Event
KS4
to()
Sink Processor
KS0
mapValues()
Masking
KS6
flatMapValues()
Analytics Record
KS7
to()
Sink Processor

```
Topologies:
   Sub-topology: 0
    Source: KSTREAM-SOURCE-0000000000 (topics: [pos])
      --> KSTREAM-FILTER-0000000003, KSTREAM-MAPVALUES-0000000006, KSTREAM-FILTER-0000000001
    Processor: KSTREAM-FILTER-0000000003 (stores: [])
      --> KSTREAM-MAPVALUES-0000000004
      <-- KSTREAM-SOURCE-0000000000
    Processor: KSTREAM-MAPVALUES-0000000006 (stores: [])
      --> KSTREAM-FLATMAPVALUES-0000000007
      <-- KSTREAM-SOURCE-0000000000
    Processor: KSTREAM-FILTER-0000000001 (stores: [])
      --> KSTREAM-SINK-0000000002
      <-- KSTREAM-SOURCE-0000000000
    Processor: KSTREAM-FLATMAPVALUES-0000000007 (stores: [])
      --> KSTREAM-SINK-0000000008
      <-- KSTREAM-MAPVALUES-0000000006
    Processor: KSTREAM-MAPVALUES-0000000004 (stores: [])
      --> KSTREAM-SINK-0000000005
      <-- KSTREAM-FILTER-0000000003
    Sink: KSTREAM-SINK-0000000002 (topic: shipment)
      <-- KSTREAM-FILTER-0000000001
    Sink: KSTREAM-SINK-0000000005 (topic: loyalty)
      <-- KSTREAM-MAPVALUES-0000000004
    Sink: KSTREAM-SINK-0000000008 (topic: hadoop-sink)
      <-- KSTREAM-FLATMAPVALUES-0000000007
```

State stores in Kafka Streams are used to maintain and store local state information for stream processing tasks. They allow processors to store and access intermediate results, aggregate data, or perform join operations efficiently.