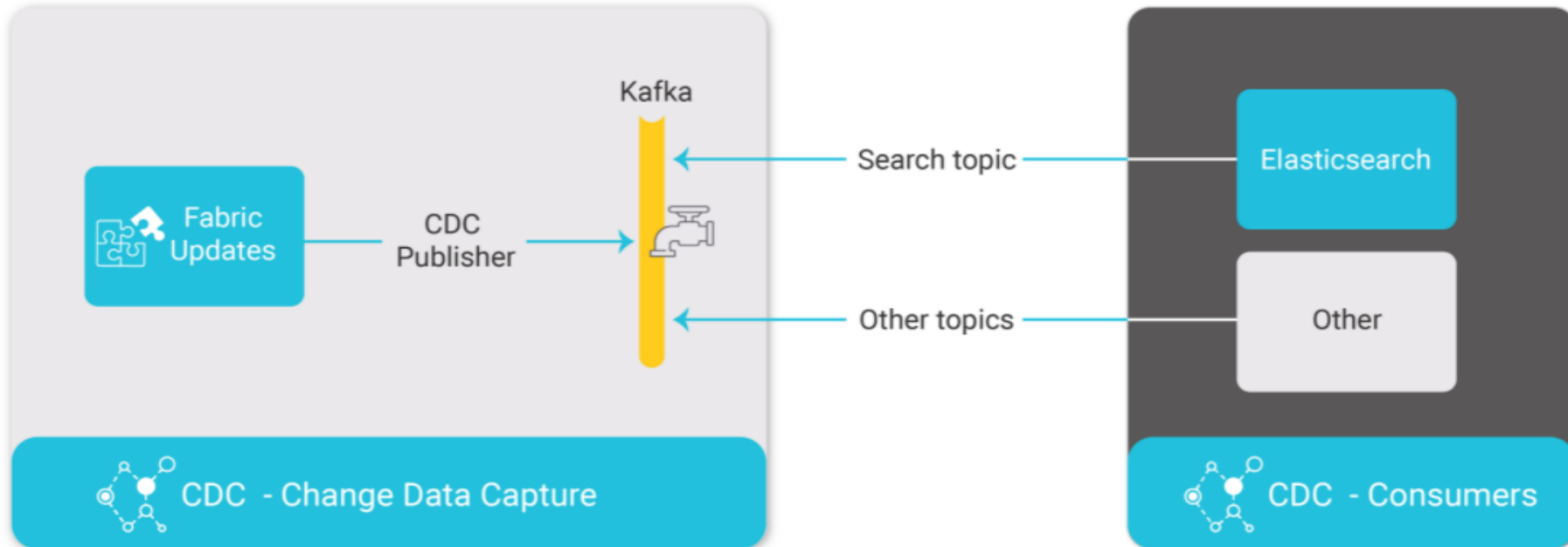


# Architecture & Implementation

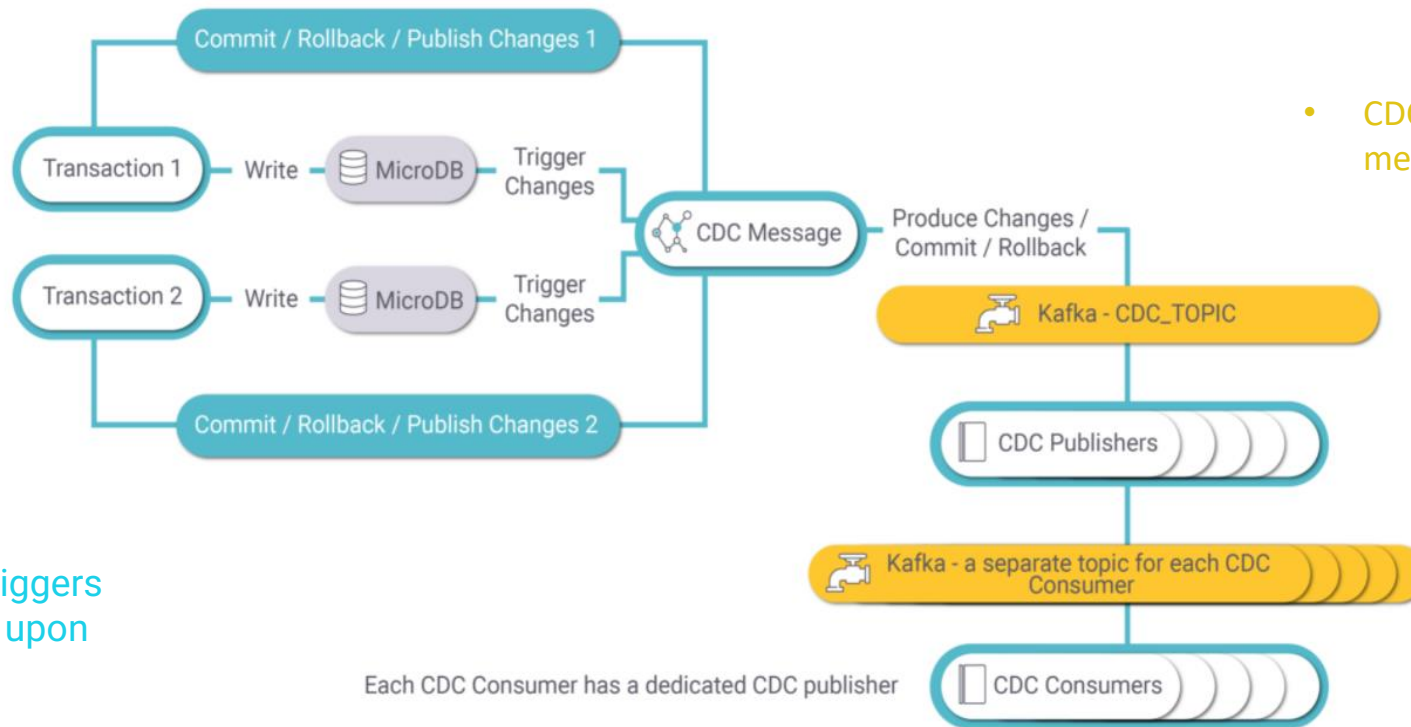


Fabric CDC\_TRANSACTION\_PUBLISHER job publishes CDC changes to Kafka.  
Each CDC consumer has its own Kafka topic.

Fabric CDC\_TRANSACTION\_CONSUMER internal job consumes the Search topic from Kafka and updates Elasticsearch.  
Other consumers must create their own consumer processes to consume Kafka CDC messages.



# CDC Flow



## Internal Mechanism

- Each microDB triggers a CDC Message upon change
- Kafka uses the CDC\_TOPIC to keep every U/I/D transaction

## External Mechanism

- Once LUI is saved
- CDC\_TRANSACTION\_PUBLISHER
  - Consumes messages from Kafka
  - Creates a CDC message for each transaction on a separate topic (one per Consumer)
- CDC « consumer » consume the messages



## CDC – Demo Flow (in video 41'00)

- Maintains subscribing systems with fresh data updated as soon as the data changes in one of the LU Instances
  - constantly tracks changes in a source database
  - immediately updates the target database
  - uses stream processing to ensure instant changes
- Flow (Check Environment)
  - Run a Kafka / Zookeeper environment
    - from windows
  - Create a topic + Produce & Subscribe terminals
    - Test topic
  - Publish/subscribe example
    - Send/receive messages



## CDC – Flows & Updates (in video 44'05'')

- Flow 2 – Publish all changes to a maintenance Salesforce tables
  - Flow CDC\_ConsumerToSalesForce
    - 1. Generate a change to a specific LU Table
    - 2. Show the change propagation via Broadway flow
      - 2 different updates from 2 different fields
    - 3. Publish to S/F



# Elastic Search – another use case for CDC (in video 49'50'')

- Elastic Search
  - Uses Kafka to enable cross-instances searches
  - Stores and processes data, typically large amounts of data
- Straightforward
  - Define E/S as an interface
  - Run Elastic Search Queries
  - Define search fields in the schema
    - **keyword**, enables a search by this column. (enables searching for an exact match case sensitive of the searched value).
    - **date**, enables a search on a date column.
    - **data**
    - **Search templates**



# Examples

- Clean all instances and execute a search
  - Clean E/S indexes cache
- Migrate x instances and start executing searches
  - From Fabric Command line

```
search ltype=Customer TABLES=ADDRESS '{ "query": { "query_string": { "fields": ["CITY"], "query": ""} } }'
```

- Different “Search Options”
  - Defined as keyword and not data to return exact matches
  - `search ltype=CUSTOMER tables=ADDRESS '{ "query": { "match_phrase" : { "CITY.keyword": { "query" : "New York" } } } }';`