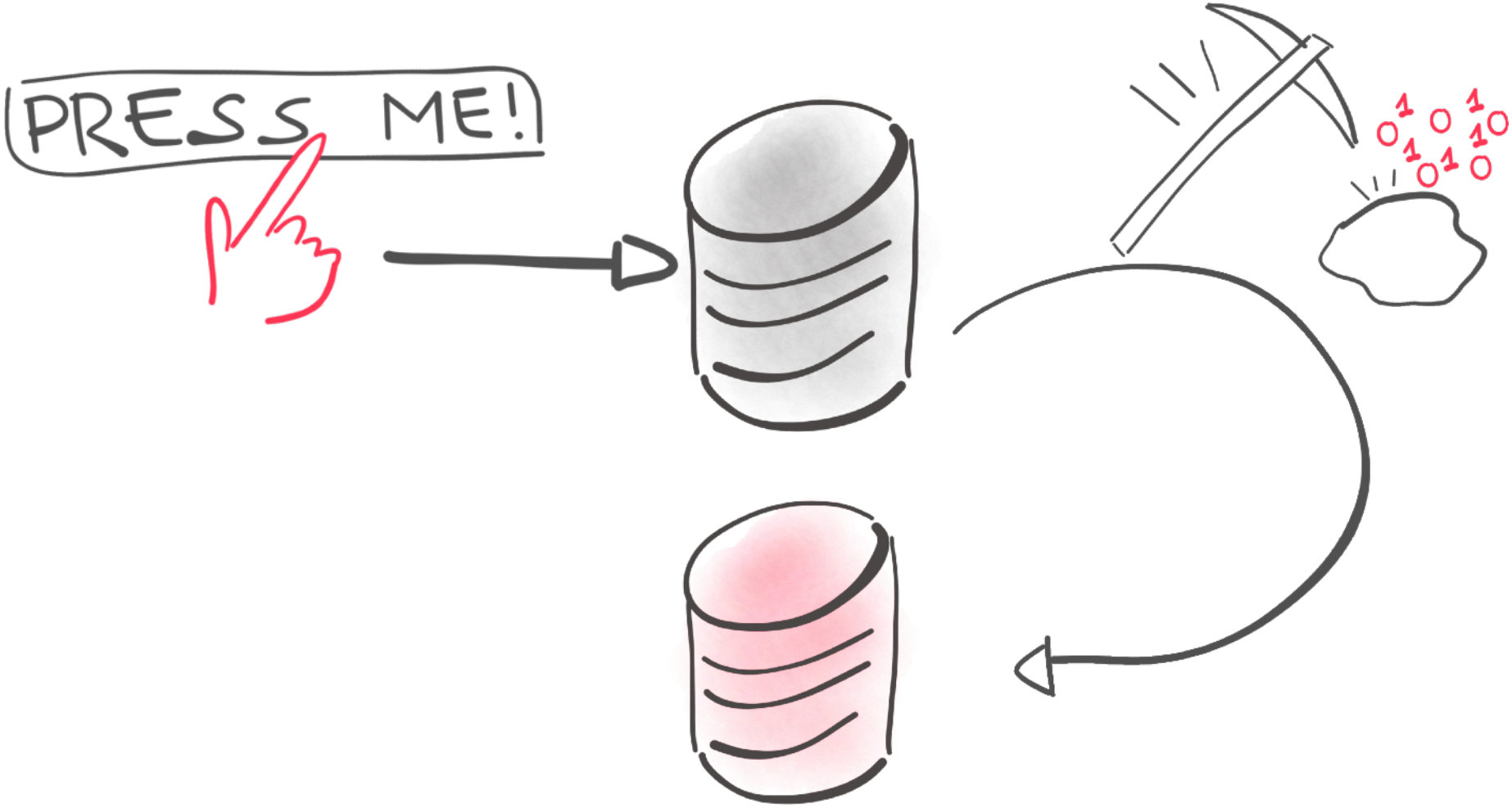
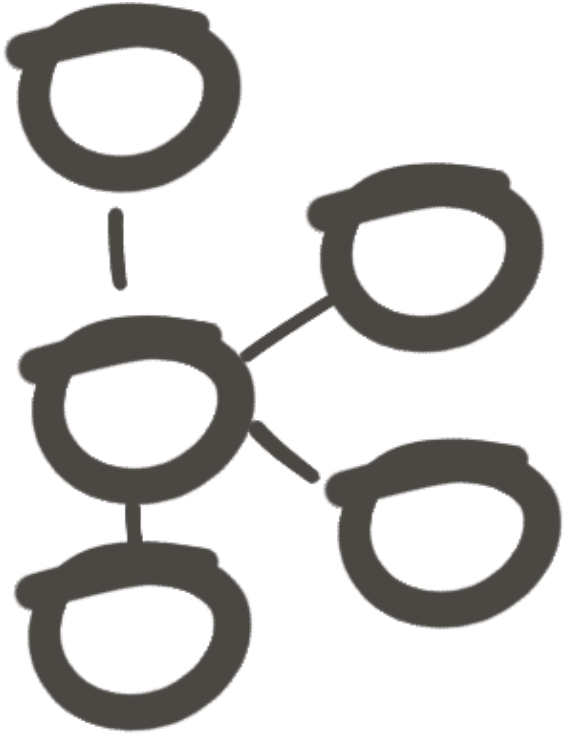
aiven

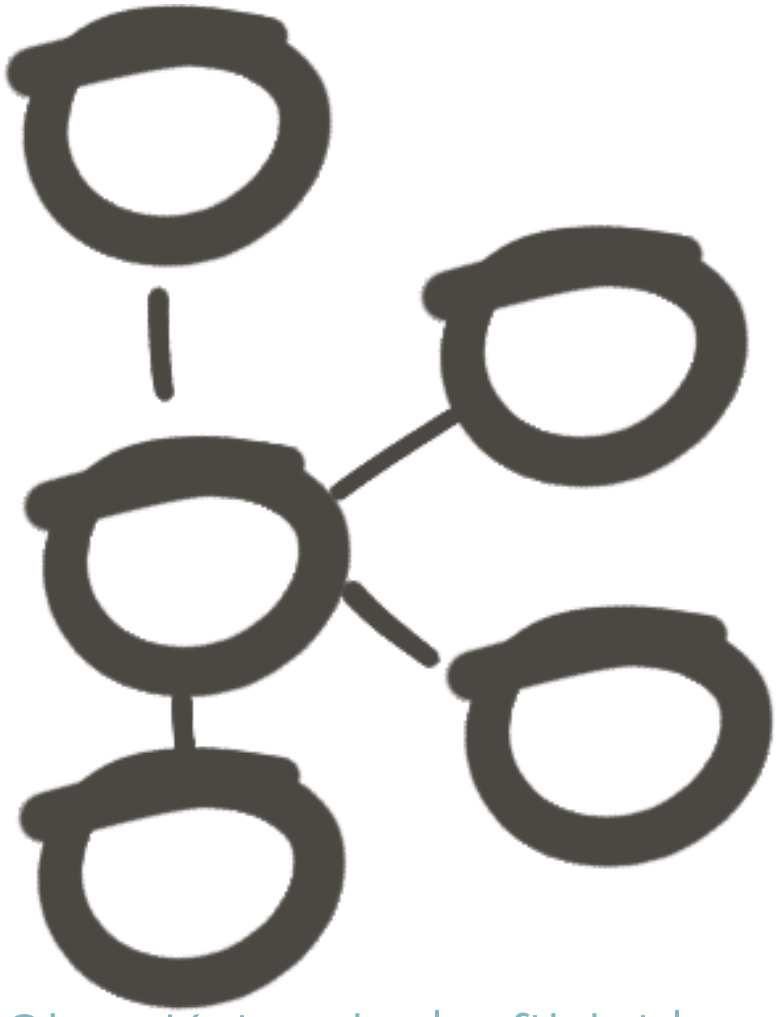
Build an Open Source Streaming Data Pipeline

Olena Kutsenko - Francesco Tisiot - Dev Advocates
@OlenaKutsenko - @ftisiot


PRESS ME!

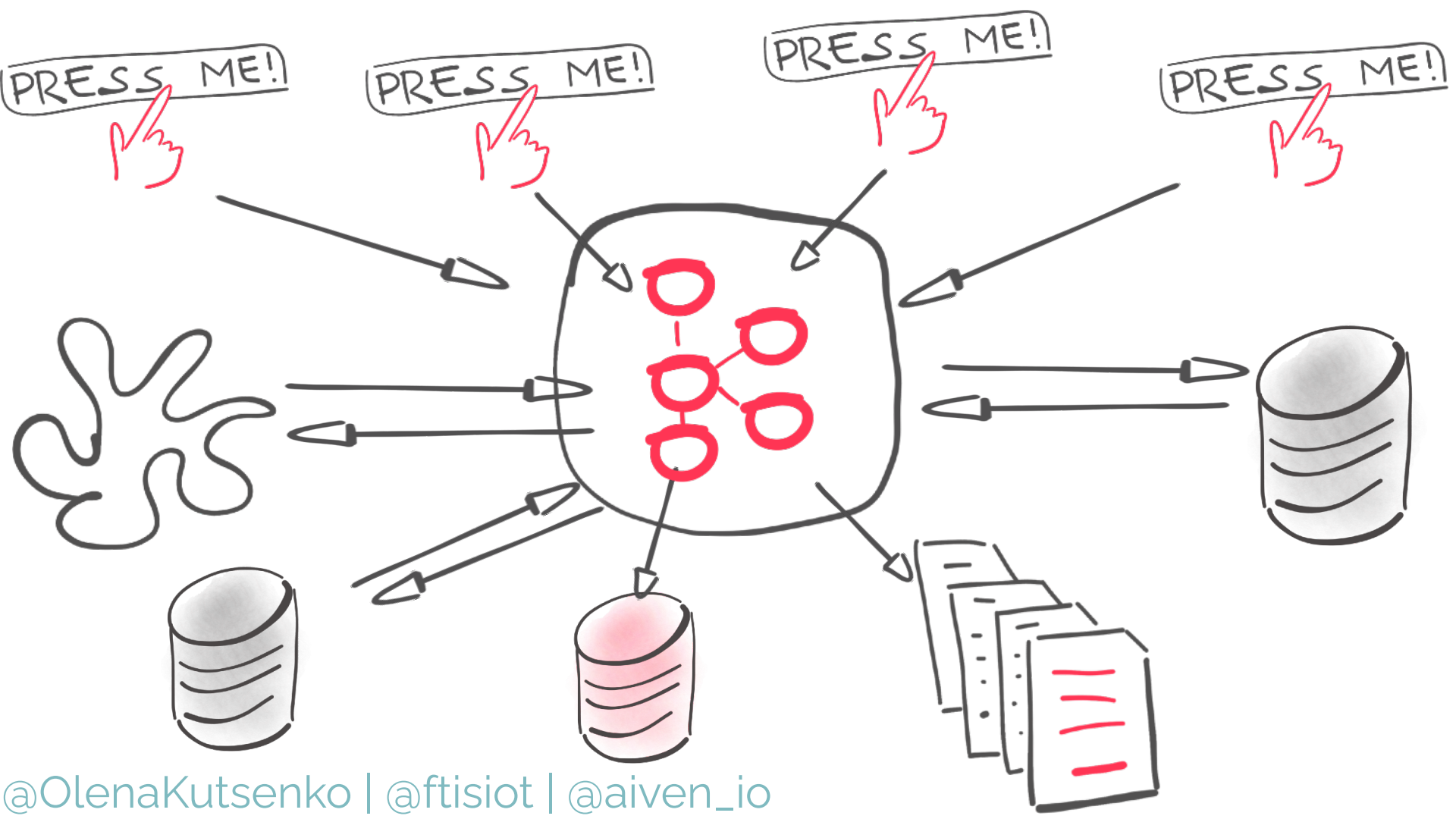

OLD

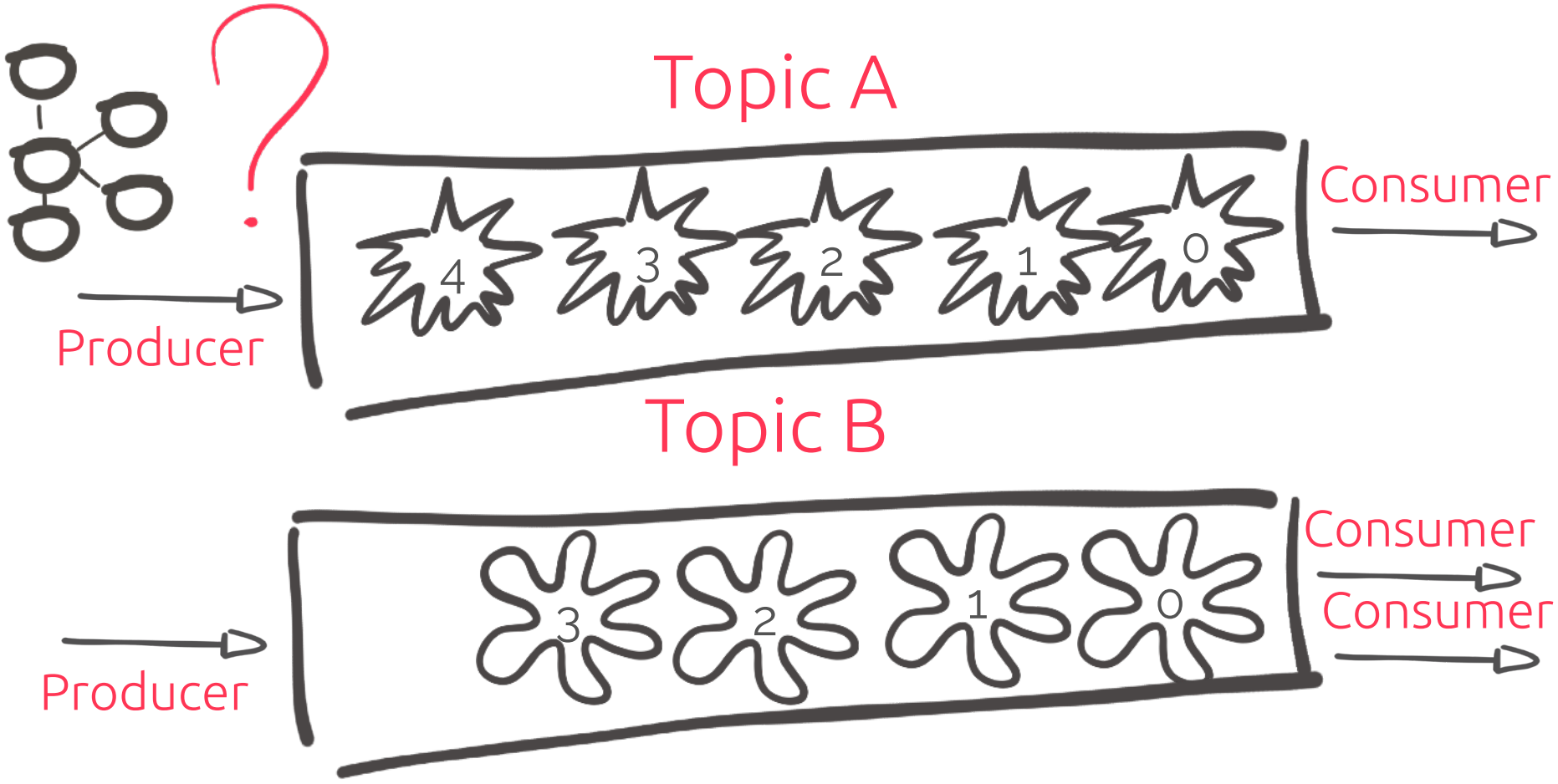


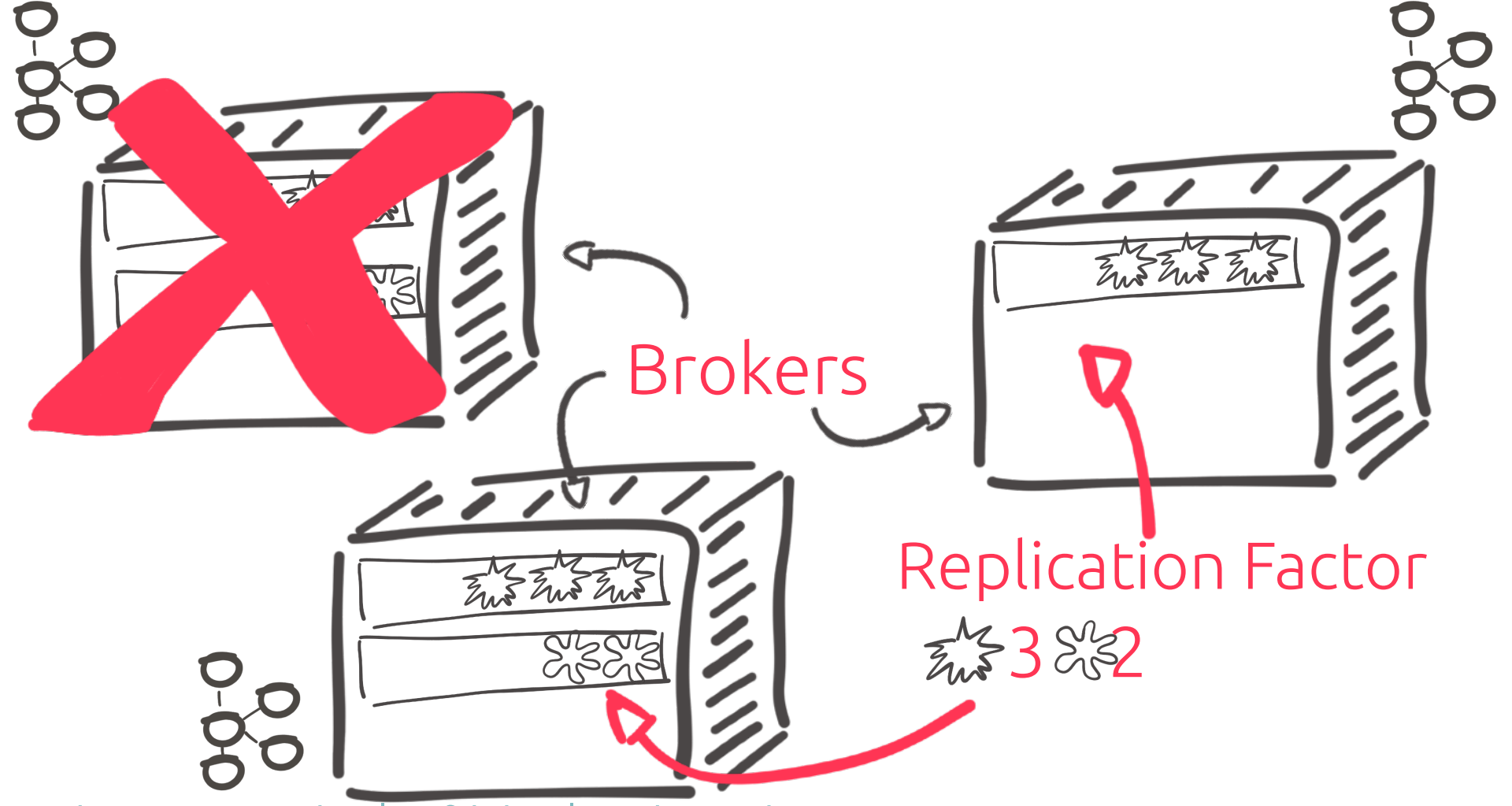


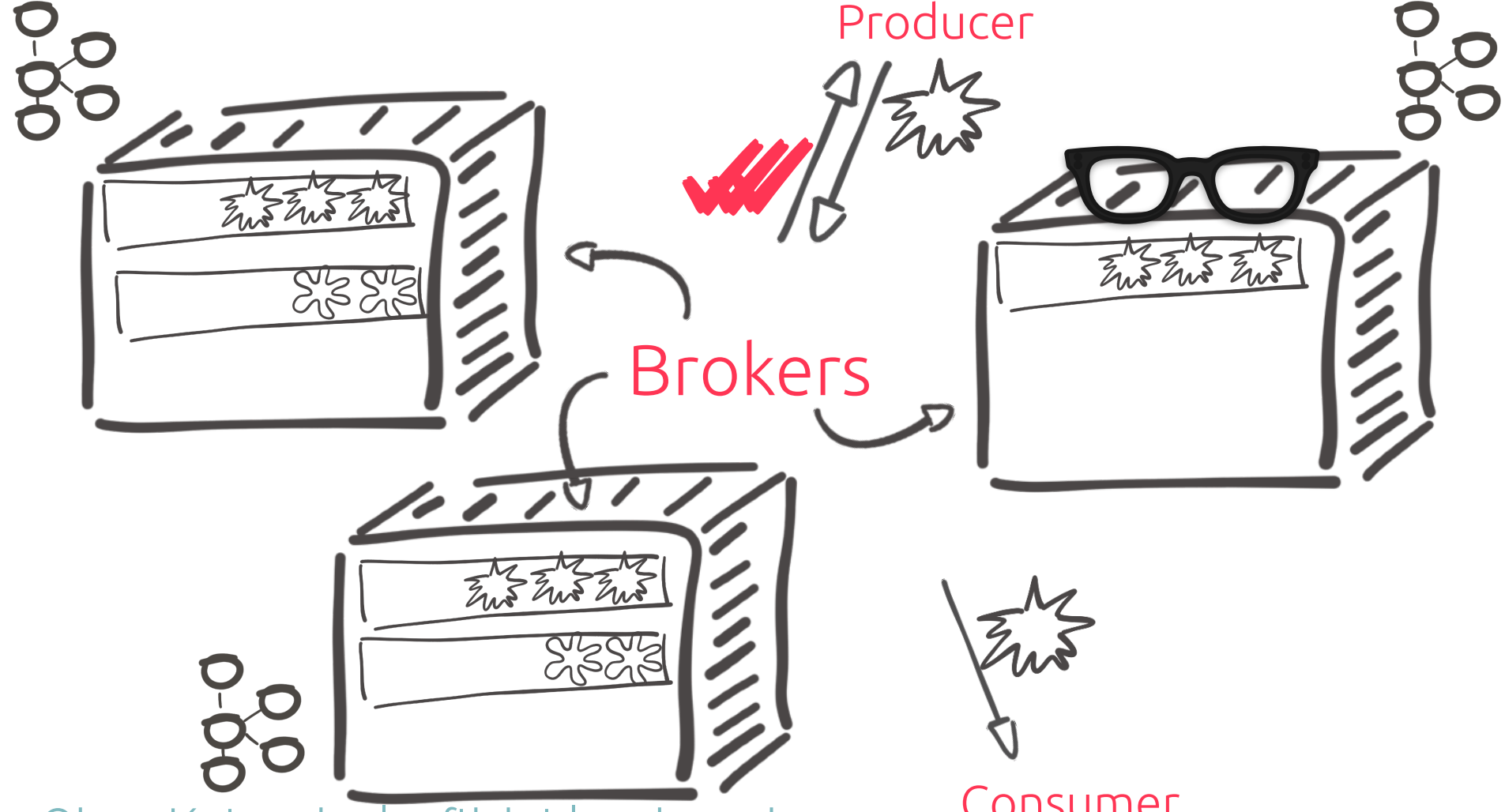
What is Apache Kafka?

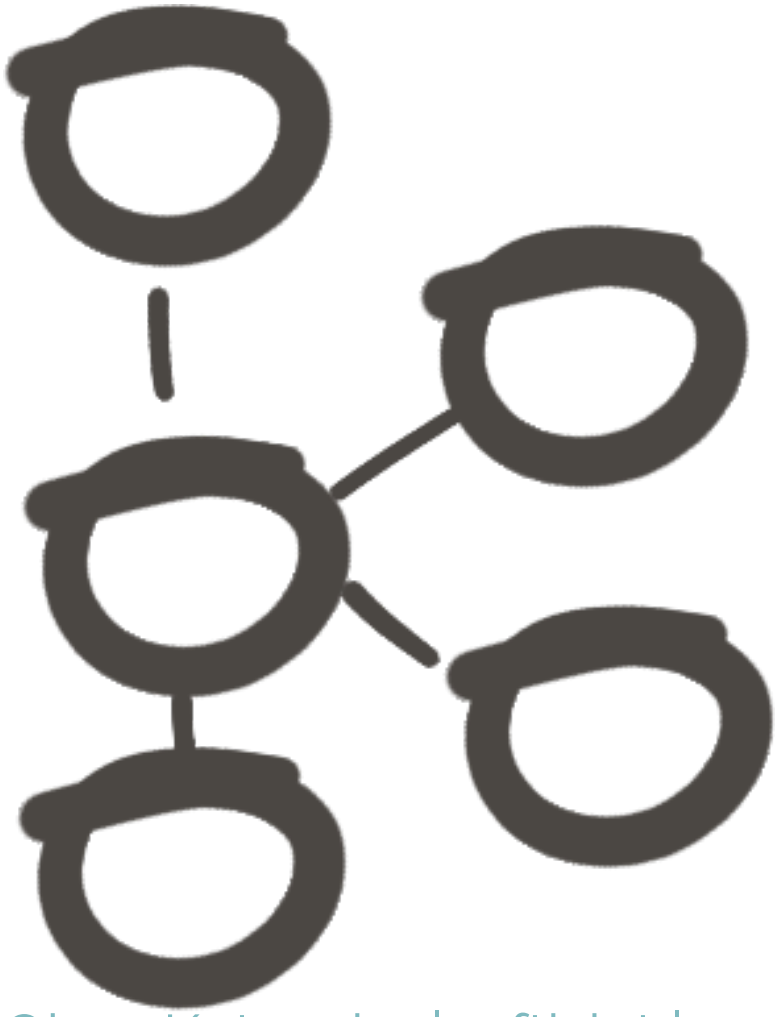
@OlenaKutsenko | @ftisiot | @aiven_io


PRESS ME!
PRESS ME!
PRESS ME!
PRESS ME!


Topic A
Consumer
4
3
2
1
0
Producer
Topic B
Consumer
Consumer
3
2
1
0
Producer

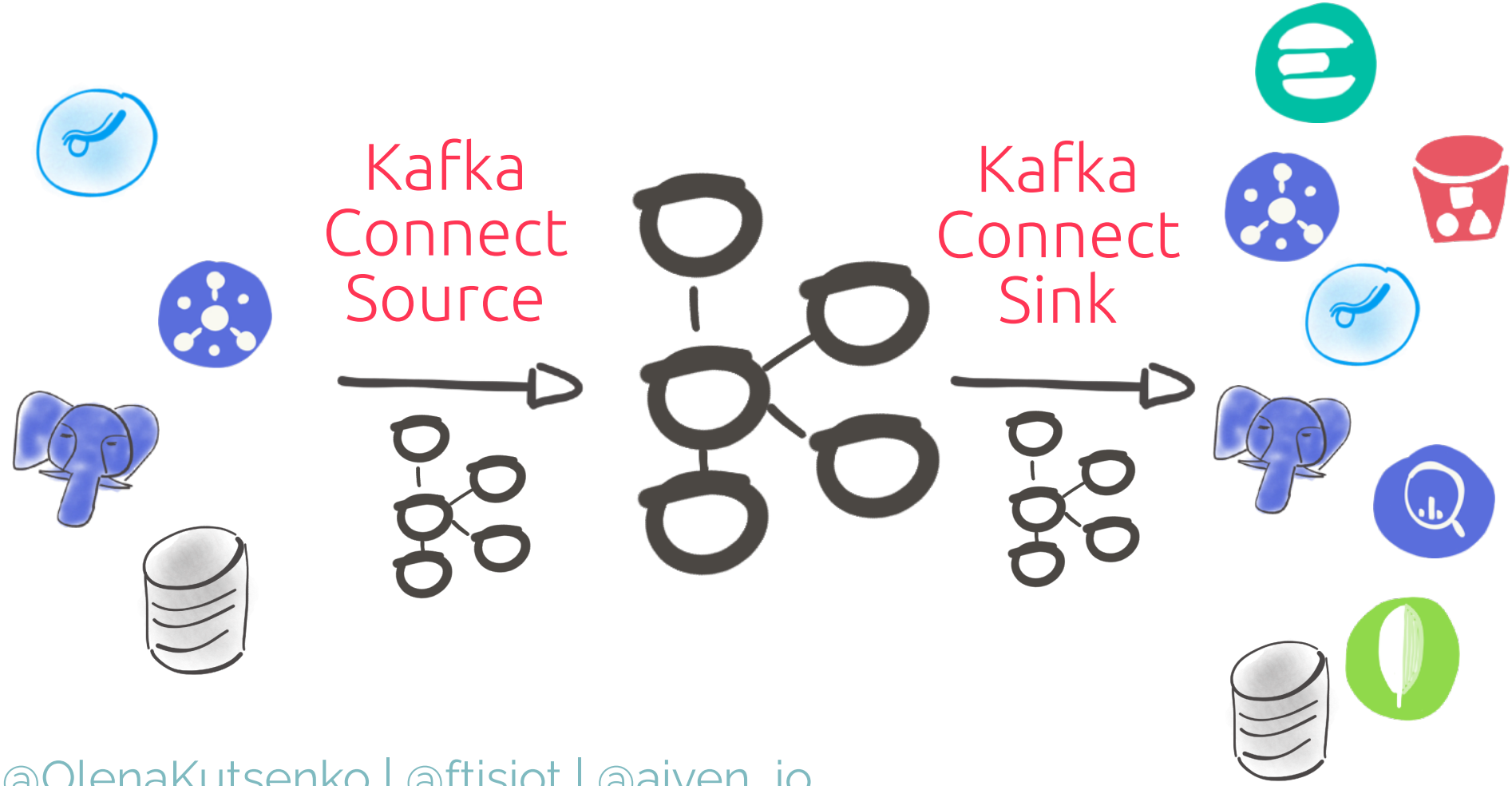

Brokers
Replication Factor
3
2


Producer
Brokers
Consumer



Integrating Apache Kafka

@OlenaKutsenko | @ftisiot | @aiven_io


Kafka
Connect
Source
Kafka
Connect
Sink

kafka-python

```
from kafka import KafkaProducer

producer = KafkaProducer(
  bootstrap_servers=['broker1:1234']
  )

producer.send(
  'my-topic-name',
  b'my-message'
  )

producer.flush()
```

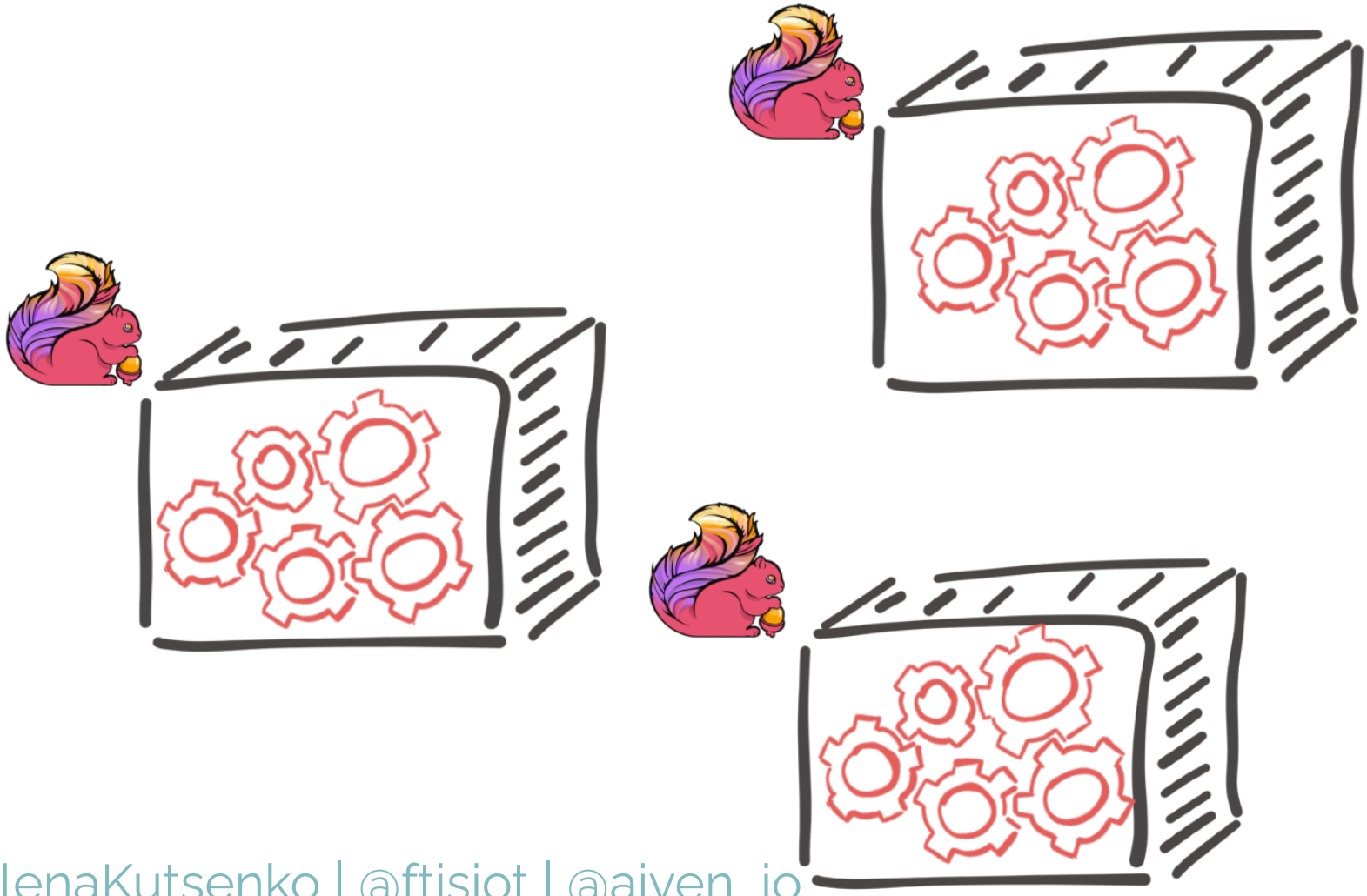
```
{
  "id": 1,
  "shop": “Mario's Pizza",
  "name": "Arsenio Pisaroni-Boccaccio",
  "phoneNumber": "+39 51 0290746",
  "address": "Via Ugo 01, Montegrotto, 85639 Padova(PD)",
  "pizzas": [
    {
      "pizzaName": "Margherita",
      "additionalToppings": ["ham"]
    },
    {
      "pizzaName": "Diavola",
      "additionalToppings": ["mozzarella","banana","onion"]
    }]
}
```

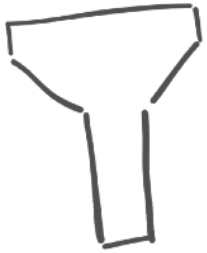
https://github.com/aiven/python-fake-data-producer-for-apache-kafka

Compute State

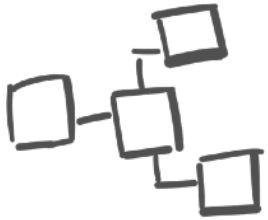
Apache Flink

@OlenaKutsenko | @ftisiot | @aiven_io





Filter

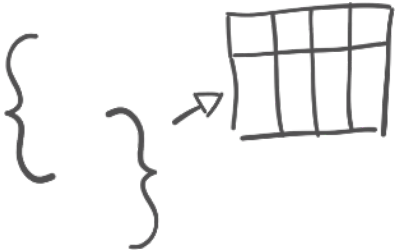


Join

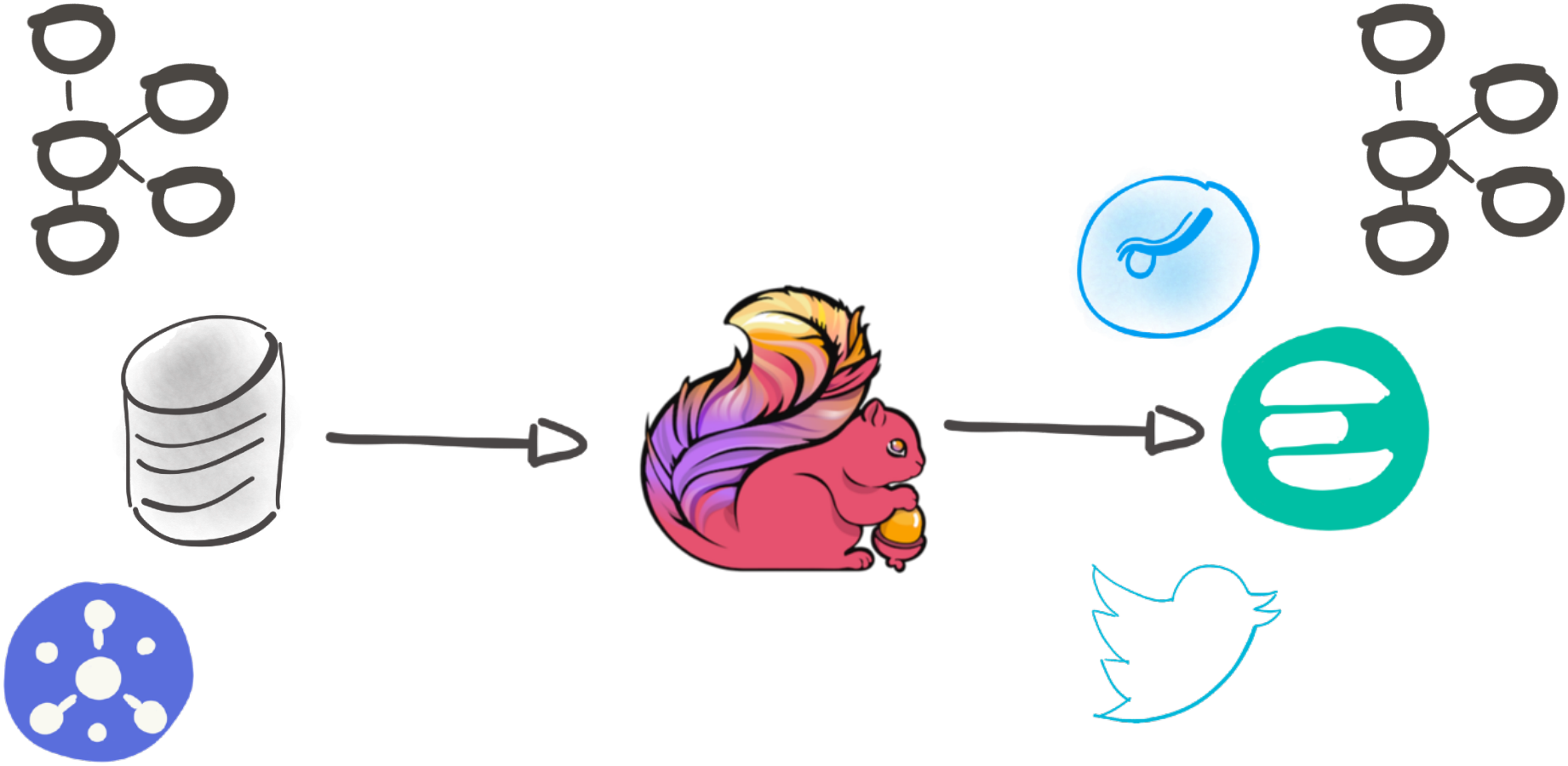
Aggregate

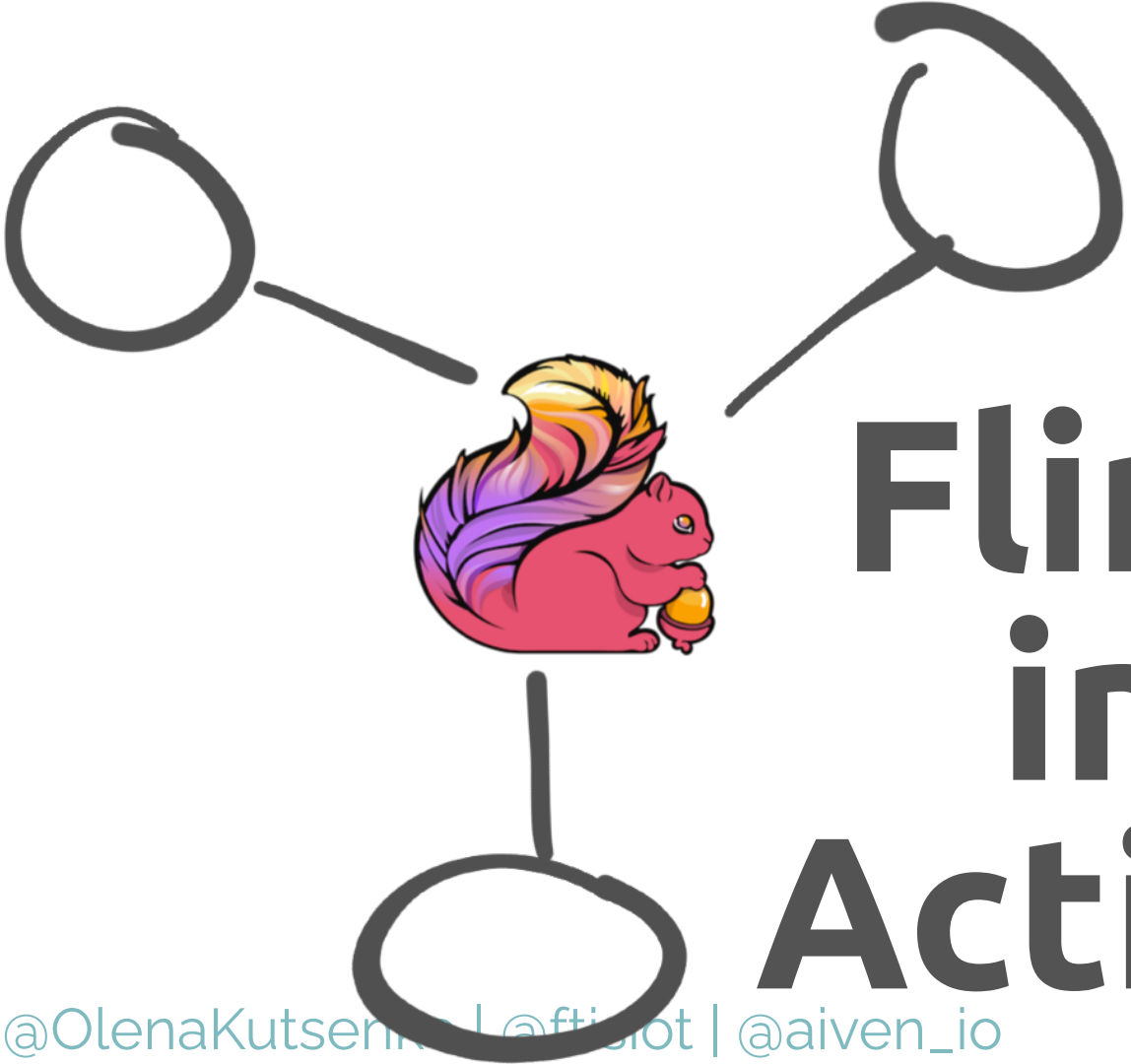
Explode

Detect

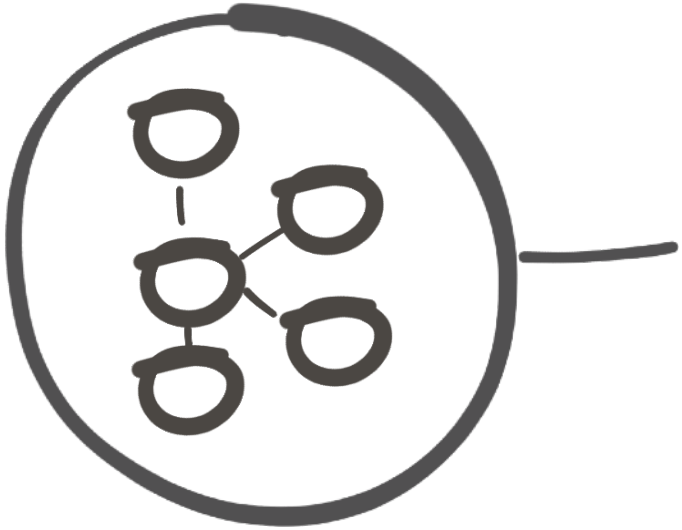


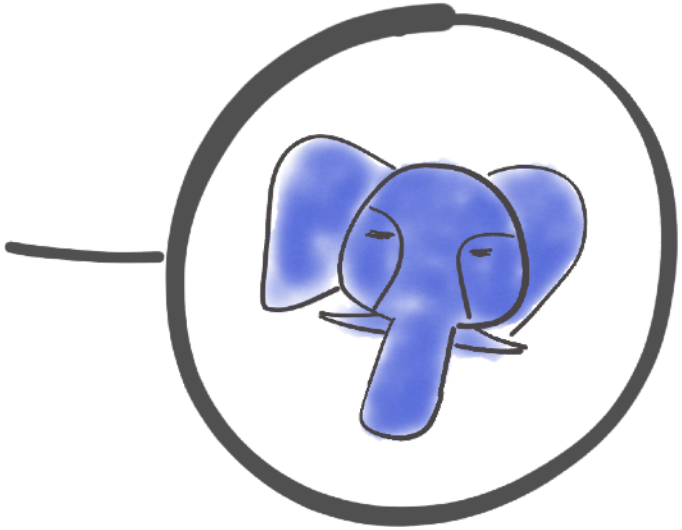
Transform





Flink in Action





```
{
  "id": 1,
  "shop": “Mario's Pizza",
  "name": "Arsenio Pisaroni-Boccaccio",
  "phoneNumber": "+39 51 0290746",
  "address": "Via Ugo 01, Montegrotto, 85639 Padova(PD)",
  "pizzas": [
    {
      "pizzaName": "Margherita",
      "additionalToppings": ["ham"]
    }]
}
```

pizza_name	base_price
Marinara	4
Diavola	6
Mari & Monti	8
Salami	7
Peperoni	8
Margherita	5

Kafka Source

```
CREATE TABLE pizza_orders (
 id INT,
 shop VARCHAR,
 name VARCHAR,
 phoneNumber VARCHAR,
 address VARCHAR,
 pizzas ARRAY
        <ROW (
            pizzaName VARCHAR,
            additionalToppings ARRAY <VARCHAR>)>
) WITH (
  'connector' = 'kafka',
  'properties.bootstrap.servers' = ‘kafka:13041',
  'topic' = 'pizza-orders',
  'scan.startup.mode' = 'earliest-offset',
```

Pg Source

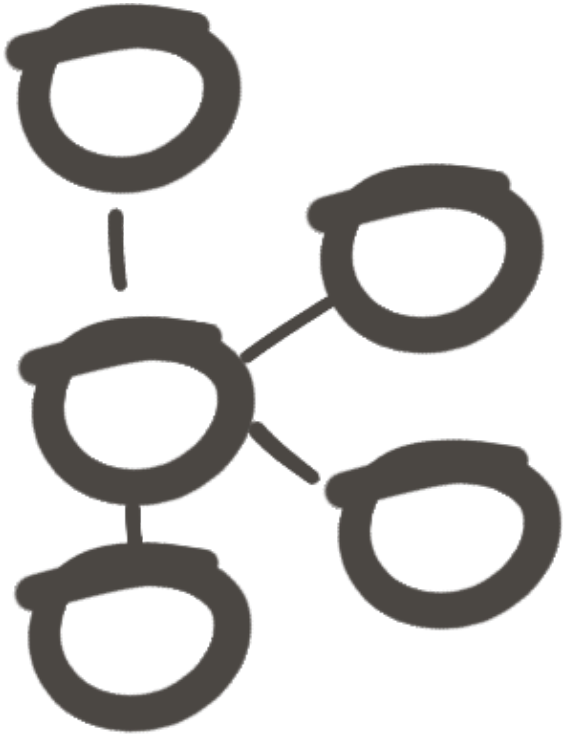
```
CREATE TEMPORARY TABLE pizza_prices (
 pizza_name VARCHAR,
 base_price INT,
 PRIMARY KEY (pizza_name) NOT ENFORCED
) WITH (
  'connector' = 'jdbc',
  'url' = ‘jdbc:postgresql:/pghost:13039/db',
  'username'='avnadmin',
  'password'='verysecurepassword123',
  'table-name' = 'pizza_price'
);
```

Pg Target

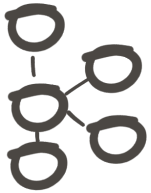
```
CREATE TABLE order_price (
 id INT,
 total_price BIGINT,
 PRIMARY KEY (id) NOT ENFORCED
) WITH (
  'connector' = 'jdbc',
  'url' = 'jdbc:postgresql://pghost:13039/db',
  'username'='avnadmin',
  'password'='verysecurepassword123',
  'table-name' = 'order_price'
);
```

Create Pipeline

```
insert into order_price
select id,
    sum(base_price) total_price
from pizza_orders cross join UNNEST(pizzas) b
LEFT OUTER JOIN pizza_prices
     FOR SYSTEM_TIME AS OF orderProctime AS pp
     ON b.pizzaName = pp.pizza_name
group by id;
```



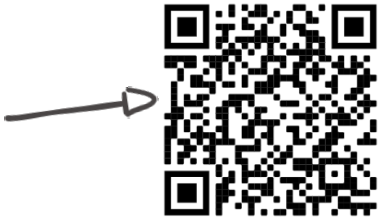
References



https://kafka.apache.org/

http://flink.apache.org/

https://github.com/aiven/sql-cli-for-apache-flink-docker



https://aiven.io/blog/create-your-own-data-stream-for-kafka-with-python-and-faker

https://aiven.io