Stream v.s. Table?

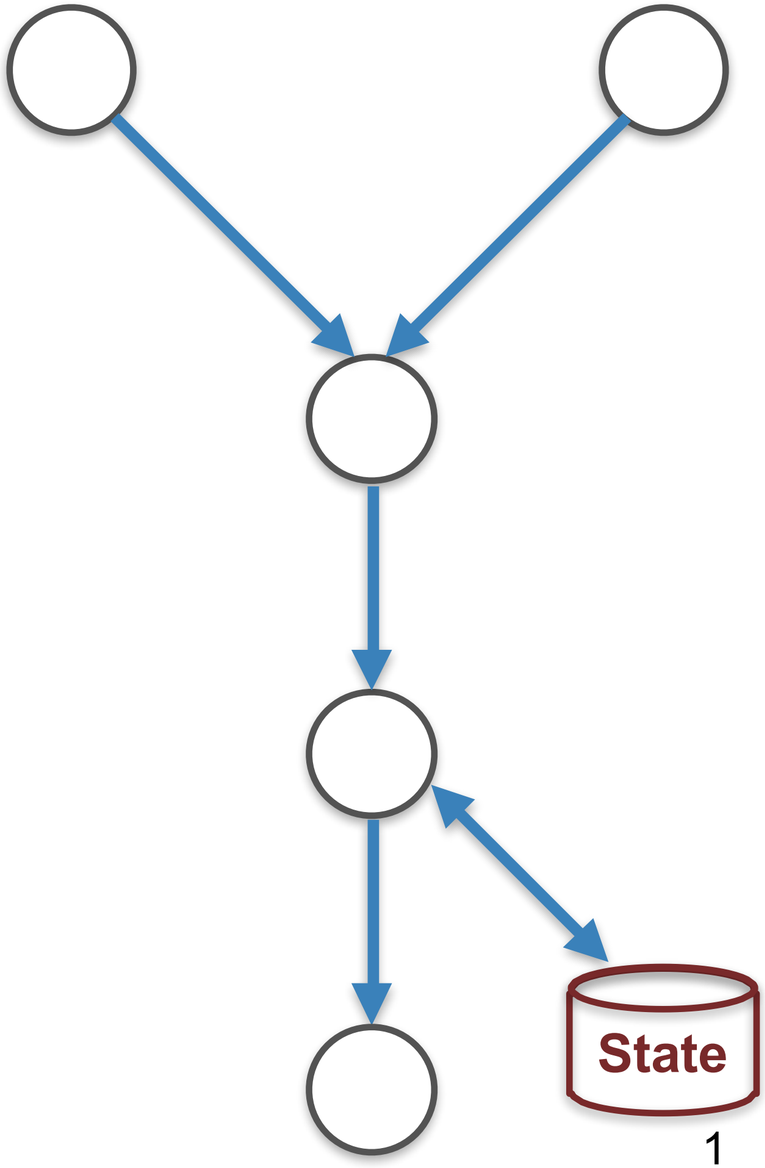
```
KStream<..> stream1 = builder.stream(”topic1”);

KStream<..> stream2 = builder.stream(”topic2”);

KStream<..> joined = stream1.leftJoin(stream2, ...);

KTable<..> aggregated = joined.aggregateByKey(...);

aggregated.to(”topic2”);
```



Tables ≈ ***Streams***

TABLES ≈ STREAMS

(key1, value1)

(key2, value2)

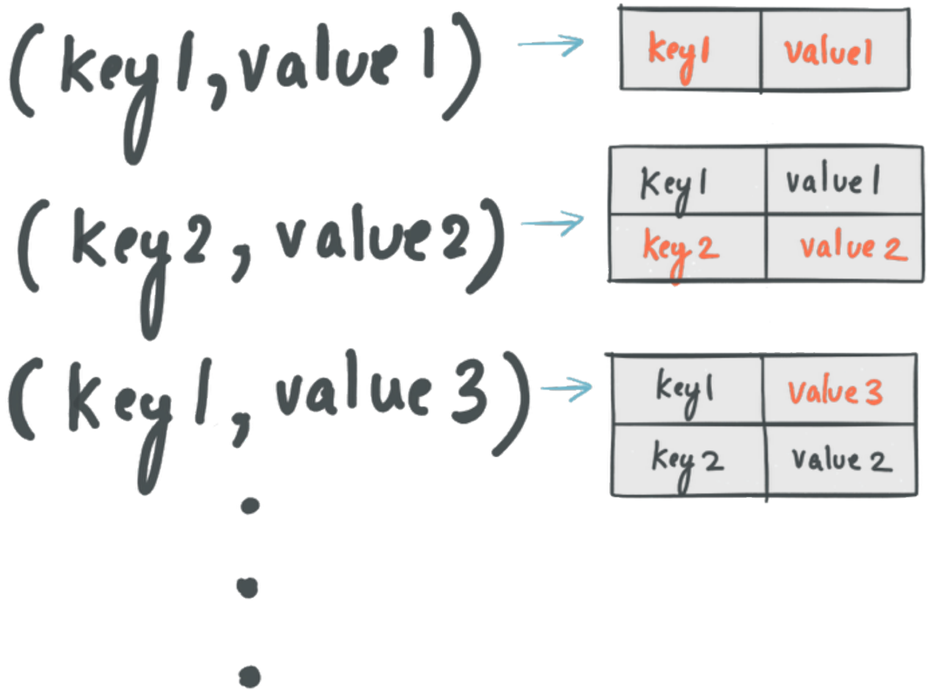
(key1, value3)

.

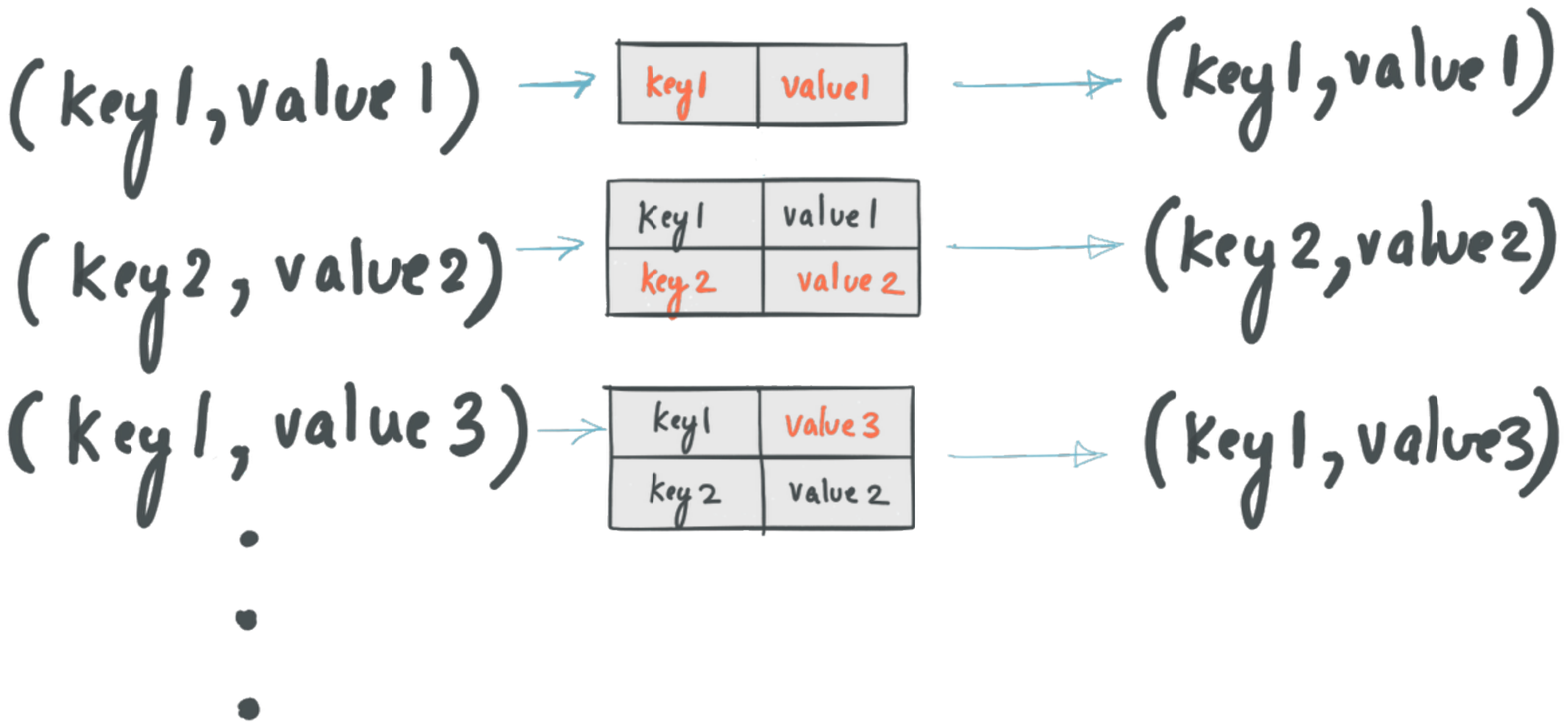
.

.

TABLES ≈ STREAMS



TABLES ≈ STREAMS



The Stream-Table Duality

- *A **stream** is a changelog of a **table***
- *A **table** is a materialized view at time of a **stream***
- *Example: change data capture (CDC) of databases*

KStream *= interprets data as record stream*

~ think: "append-only"

KTable *= data as changelog stream*

~ continuously updated materialized view

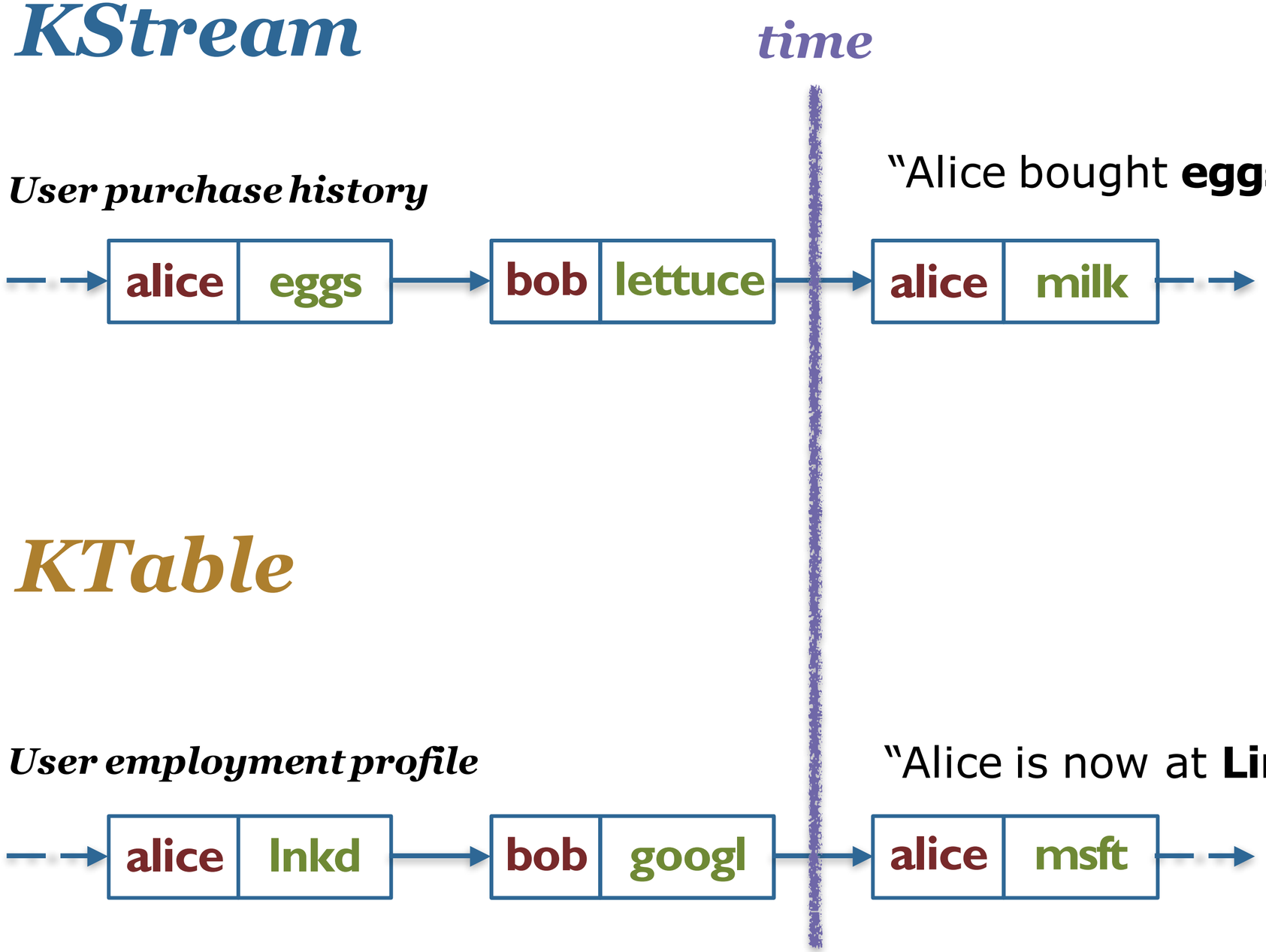
KStream

User purchase history

KTable

User employment profile

KStream



KTable

KStream

time

User purchase history

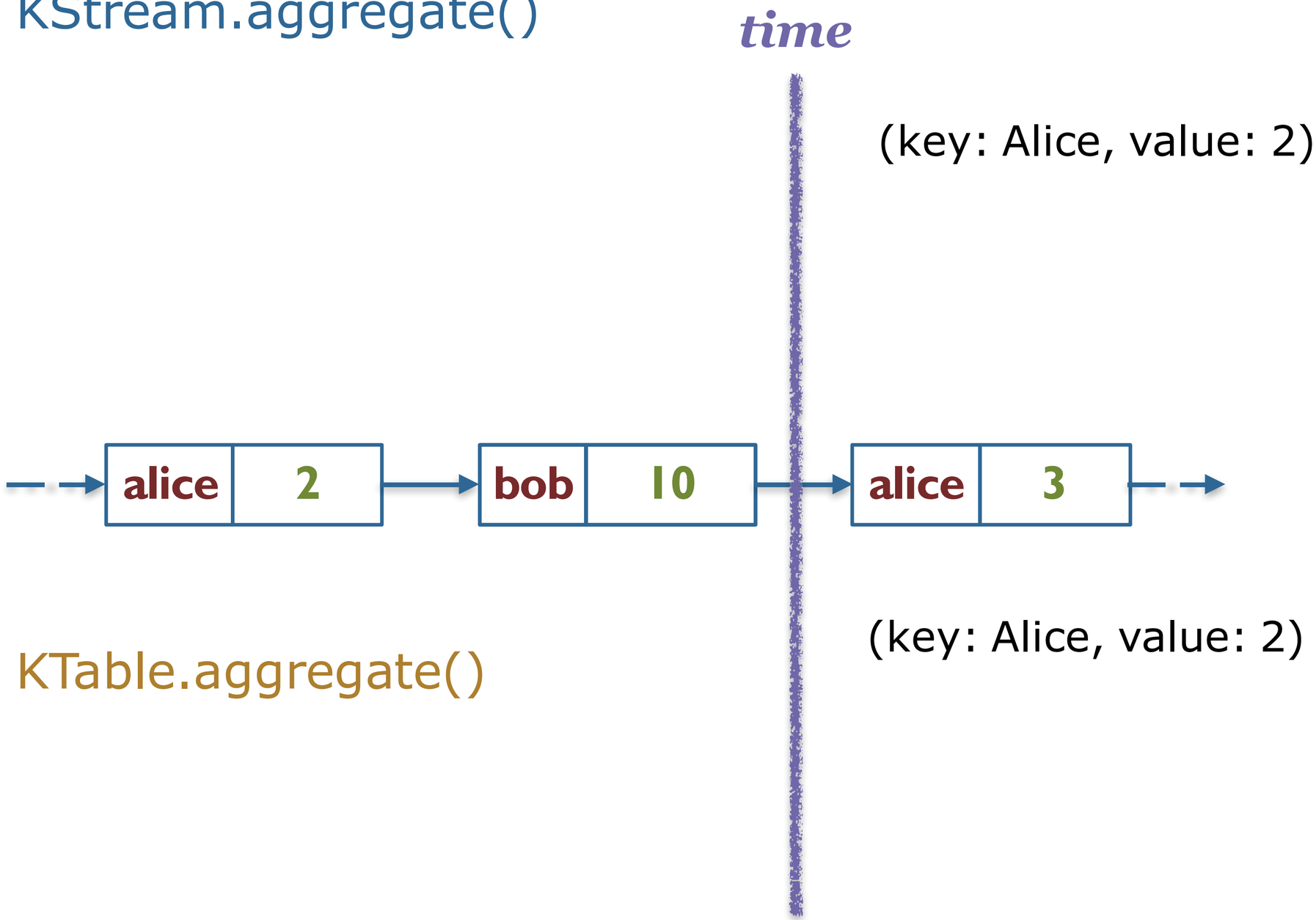
"Alice bought **eggs** and **milk**."

KTable

User employment profile

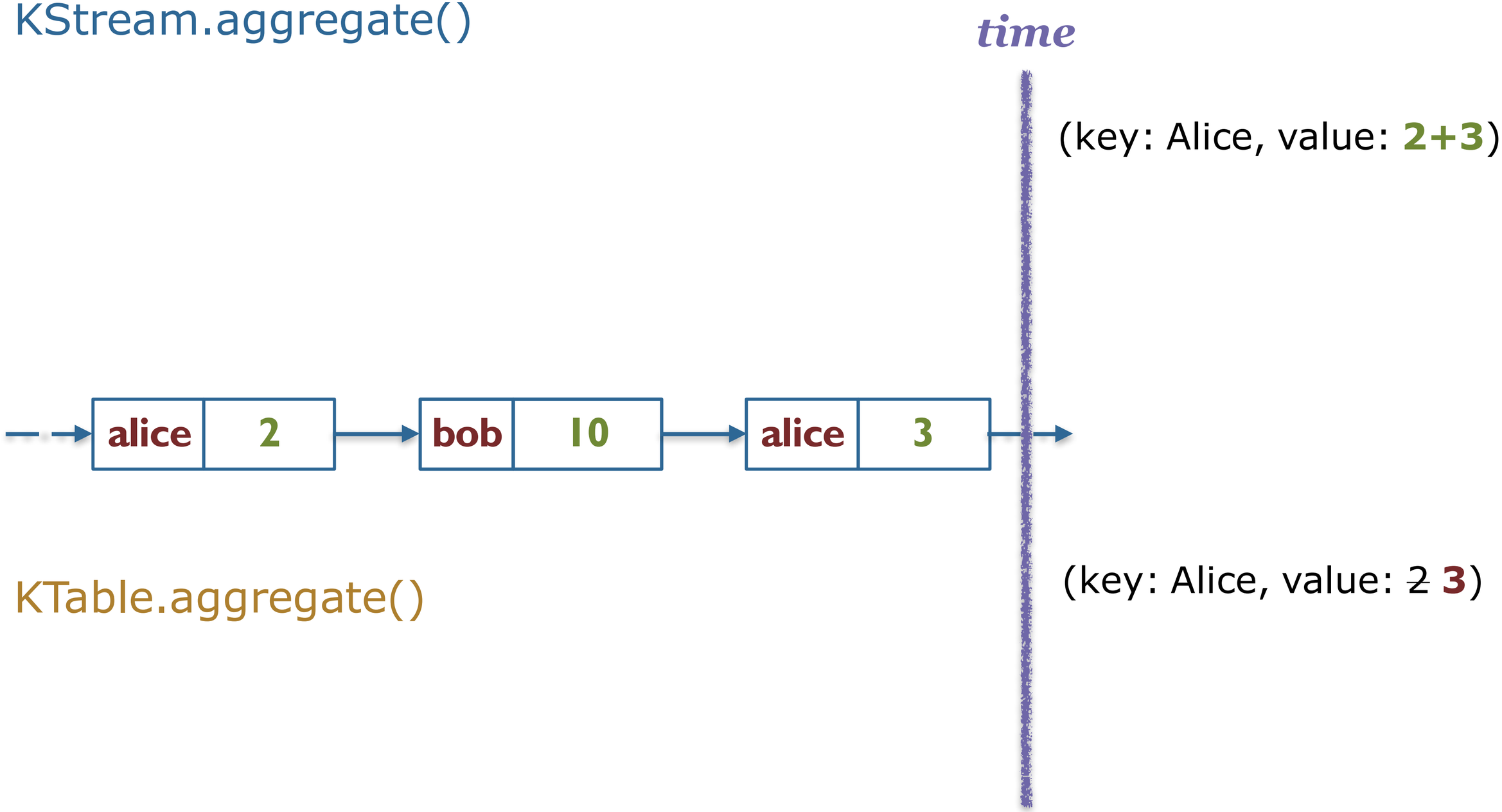
"Alice is now at ~~LinkedIn~~ **Microsoft**."

KStream.aggregate()

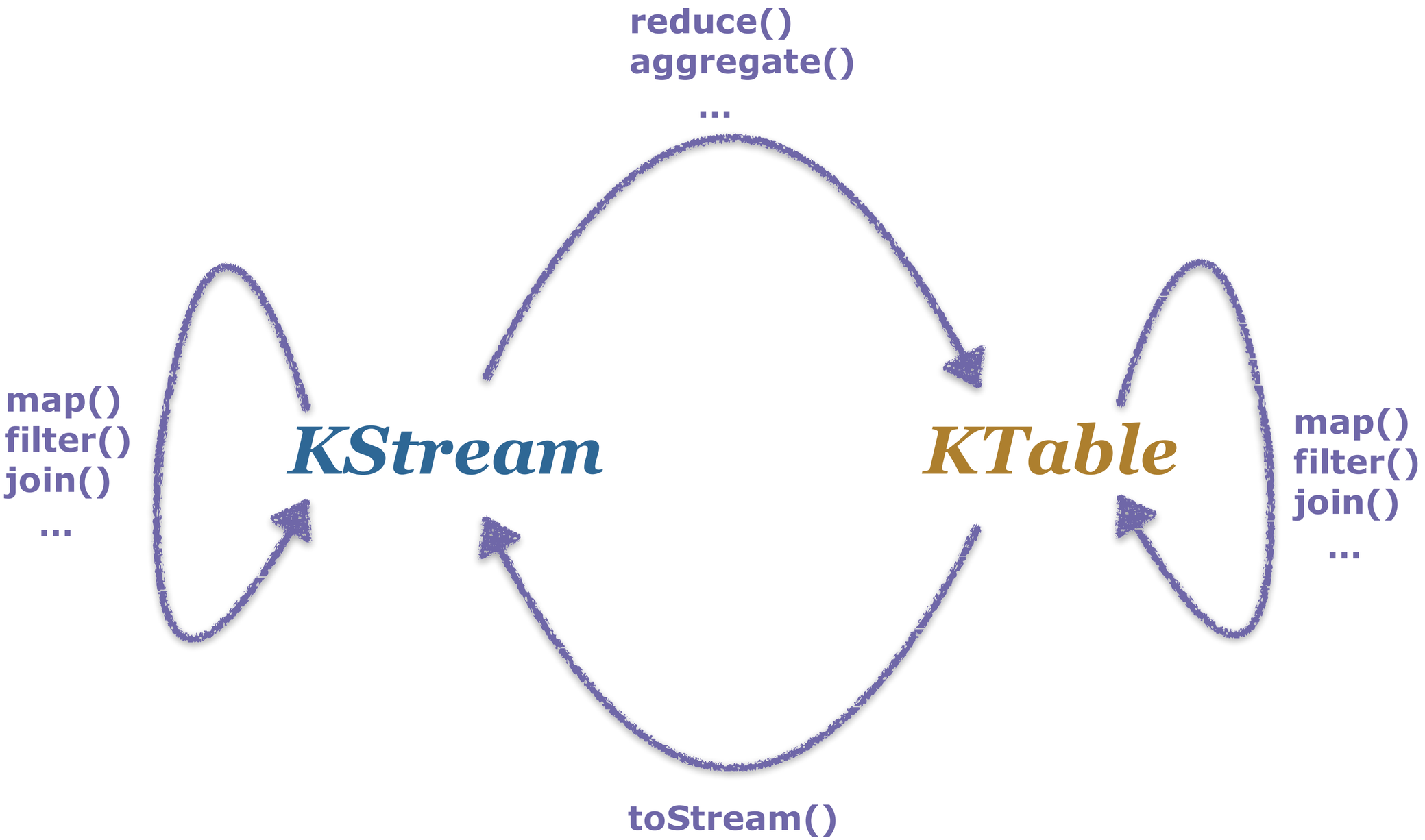


KTable.aggregate()

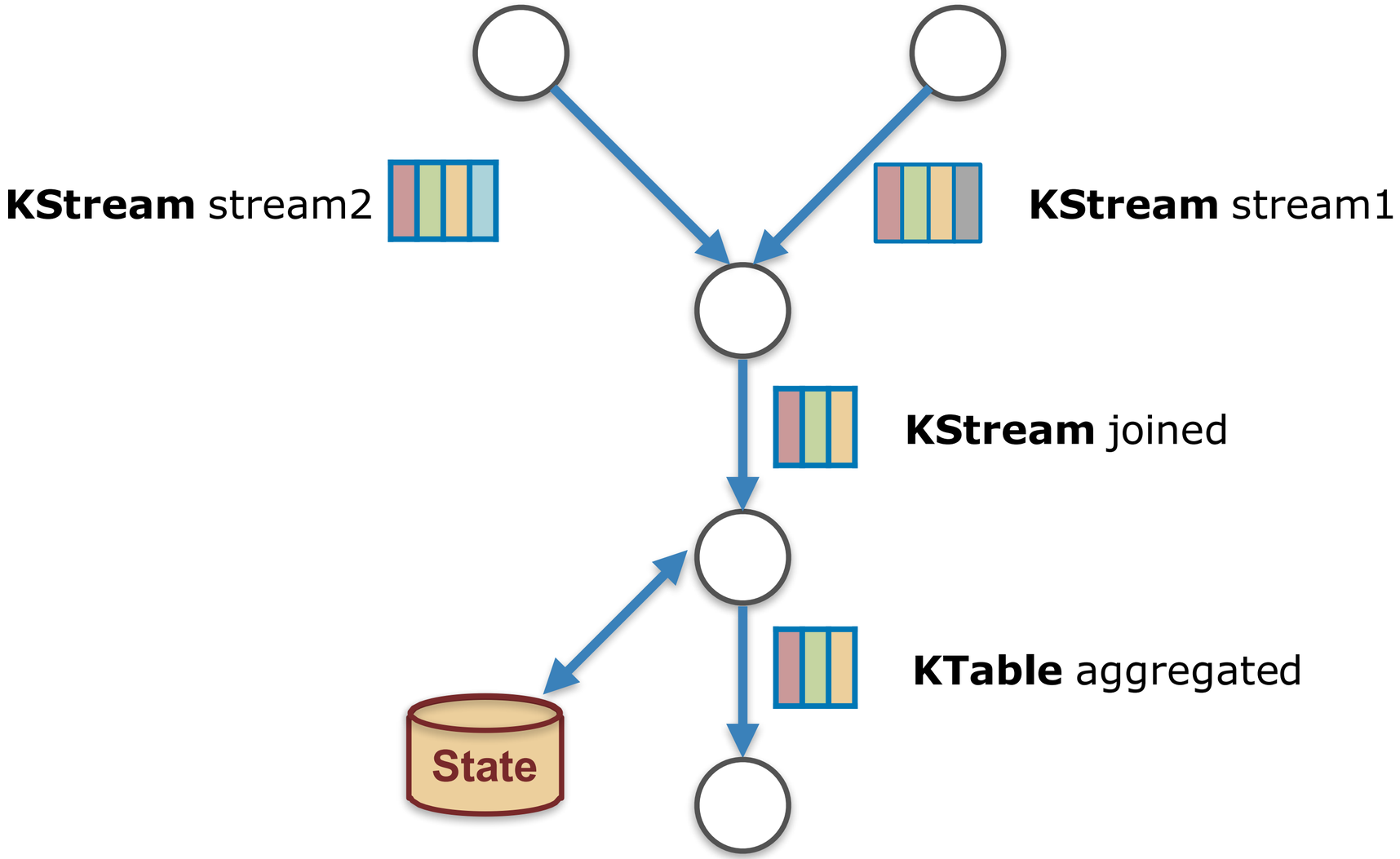
KStream.aggregate()



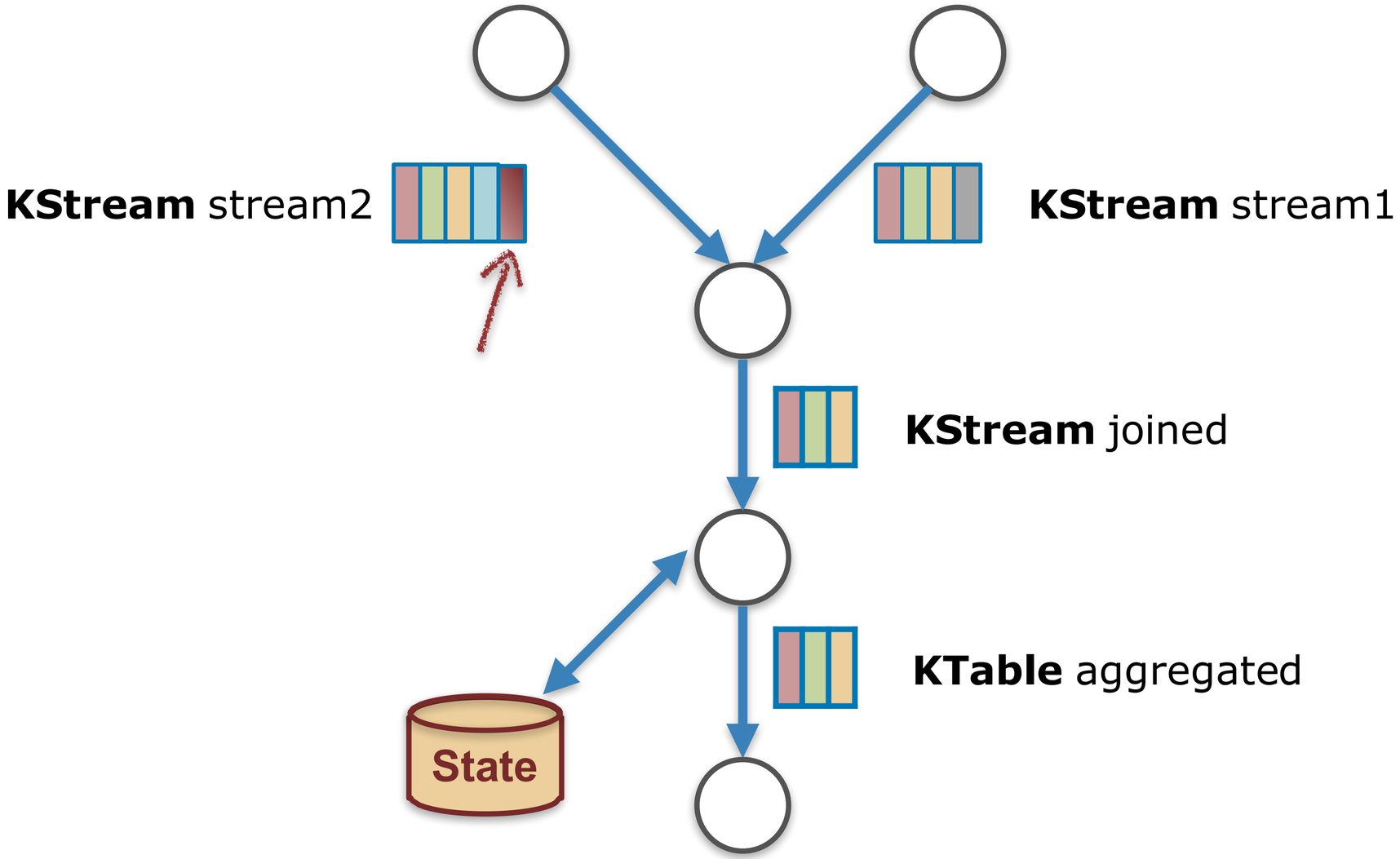
KTable.aggregate()


reduce()
aggregate()
...
map()
filter()
join()
...
KStream
KTable
map()
filter()
join()
...
toStream()

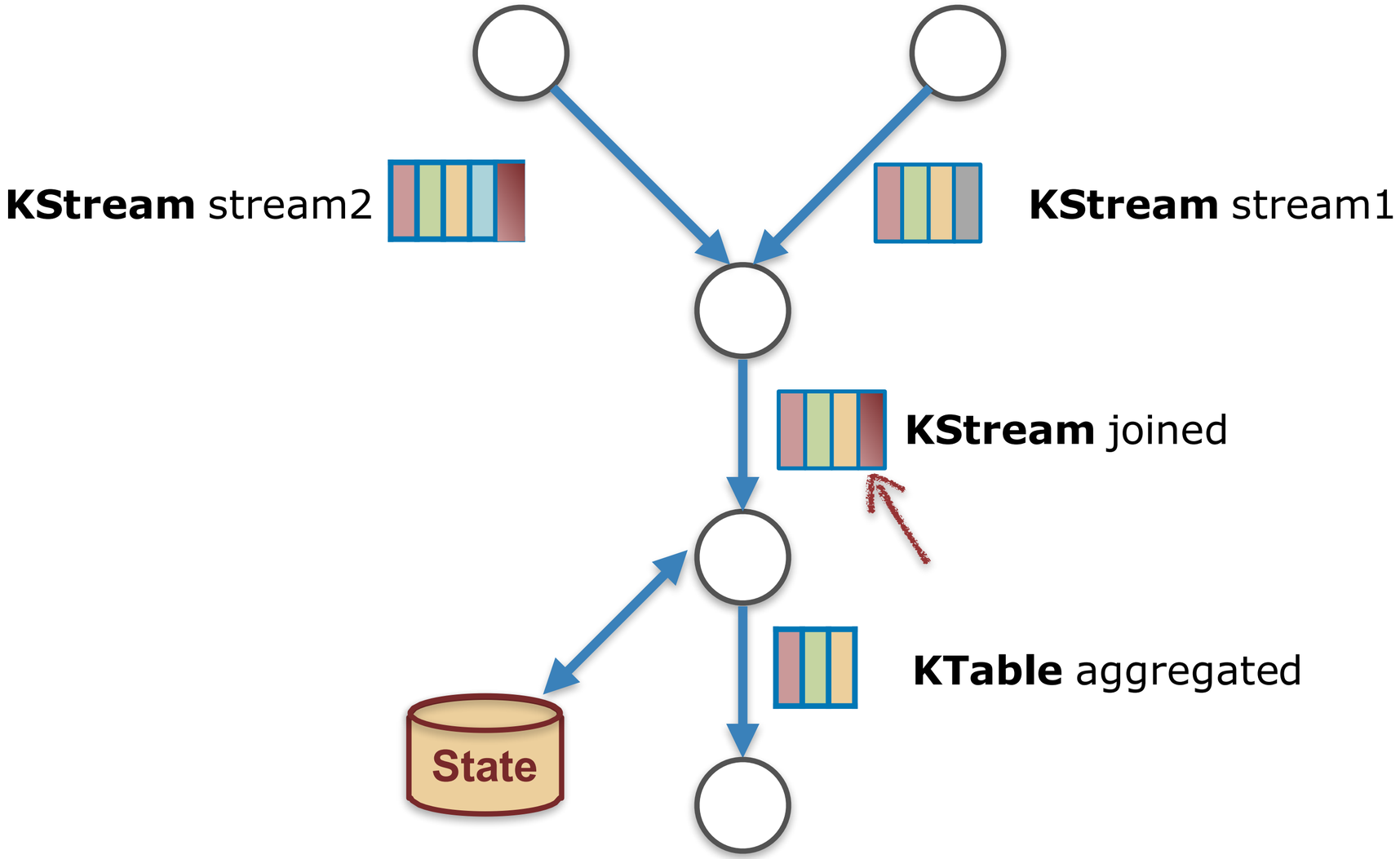
Updates Propagation in KTable



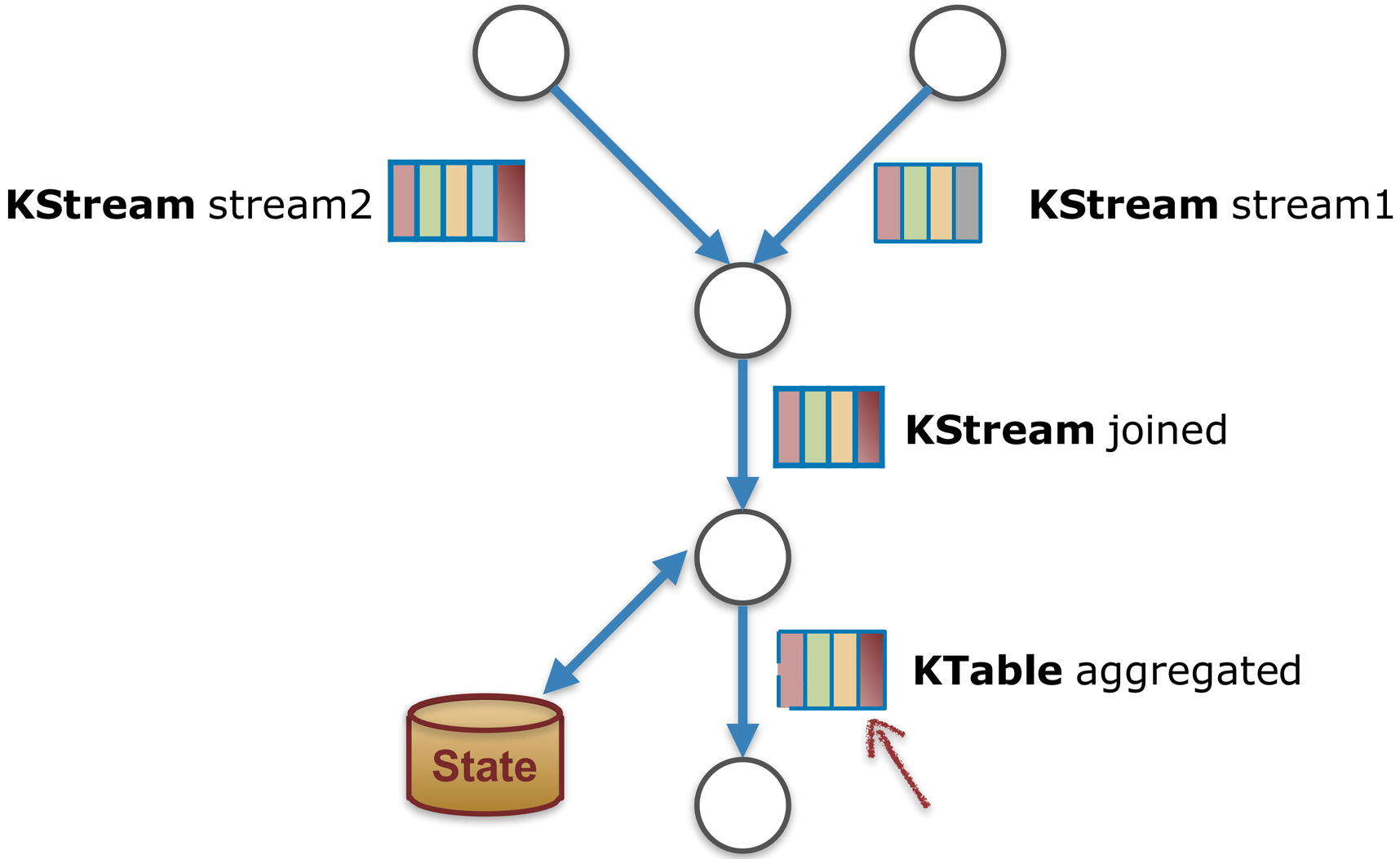
Updates Propagation in KTable



Updates Propagation in KTable



Updates Propagation in KTable



Lab: Running your first Kafka Streams Application: WordCount – 60 minutes