
Event streaming with Kafka Streams

Bobo Häggström



- Full-stack developer
- Functional programming
 - Haskell, Elm, Elixir, Reason/ReScript, Scala
- Event streaming

Event streaming

- Opposite of REST
- Tell, don't ask
- Stateful
- History of the state
- Re-use of old-data

Kafka - a short recap

- Event streaming platform
 - Pub/Sub
 - Topics
 - Key/Value
- High availability
- Scalable
- High throughput
- Permanent storage

Kafka Connect

- Framework for connecting external systems to Kafka
- Databases, message queues, FTP, etc
- Source and sink

Kafka Streams

- Java library for stream processing applications and service
- DSL
- Scalable and fault tolerant
- Loose coupling

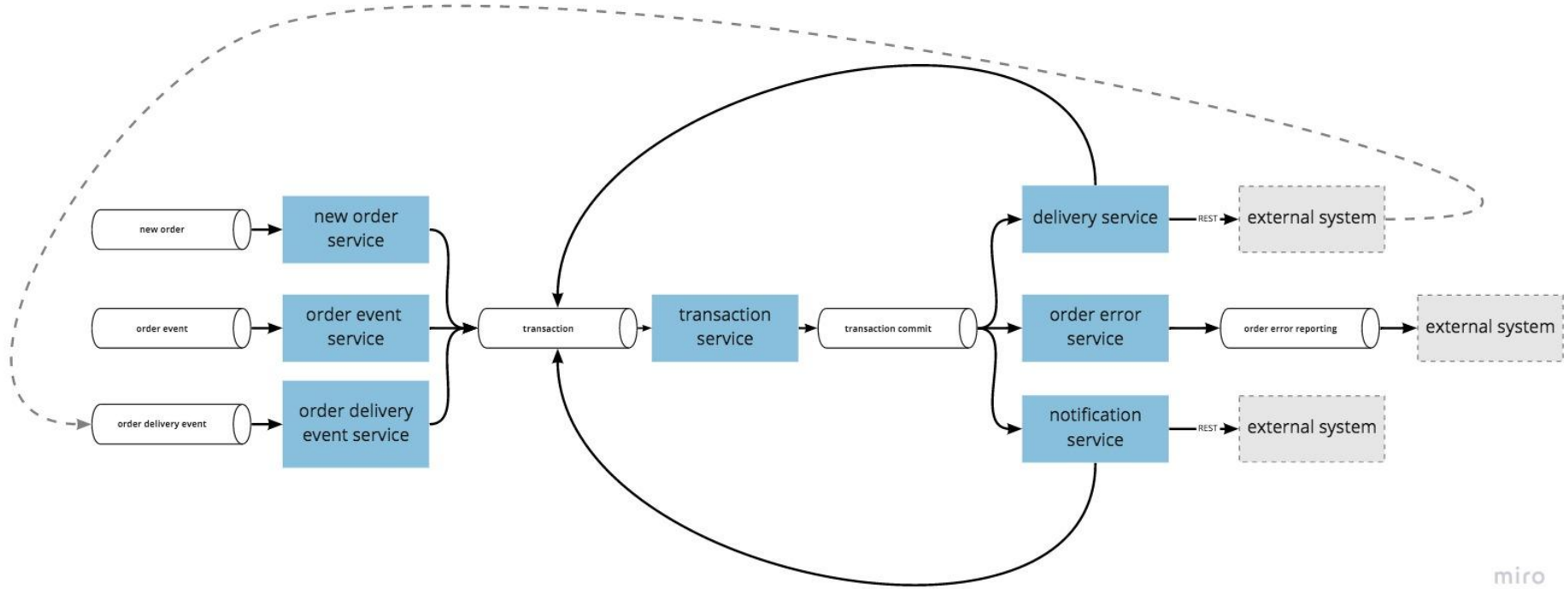
Structuring your streams

- Topic naming convention
 - *examples: <domain>-<entity/event type>, <domain>-<entity/event type>-<version>*
- Partitioning
- Data format and schema
- Single writer principle

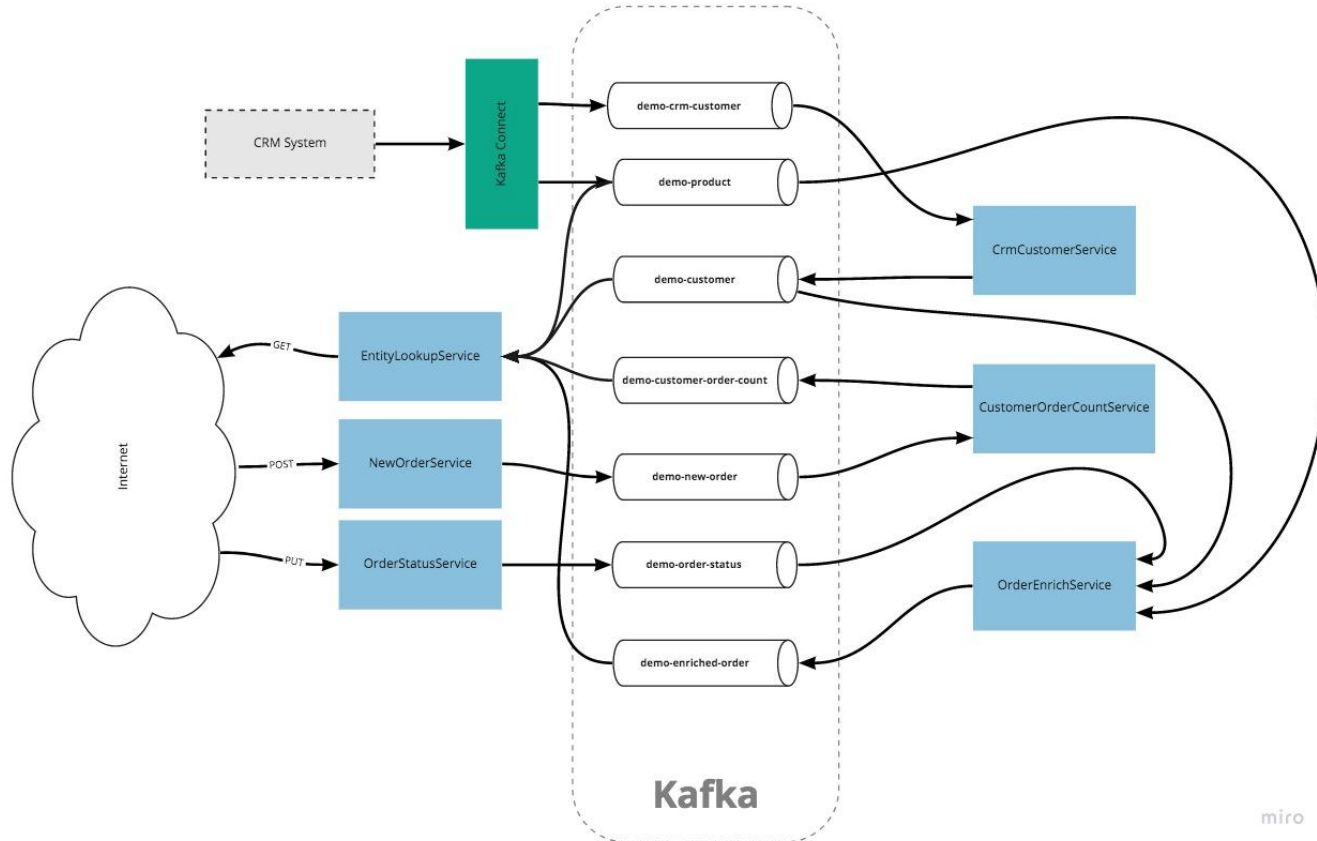
Kafka in production

- Three clusters (test, stage, prod)
- Kafka for system integration
- Some true event streaming systems
- Currently around 150 topics
- Some topics with ~2 million messages per day

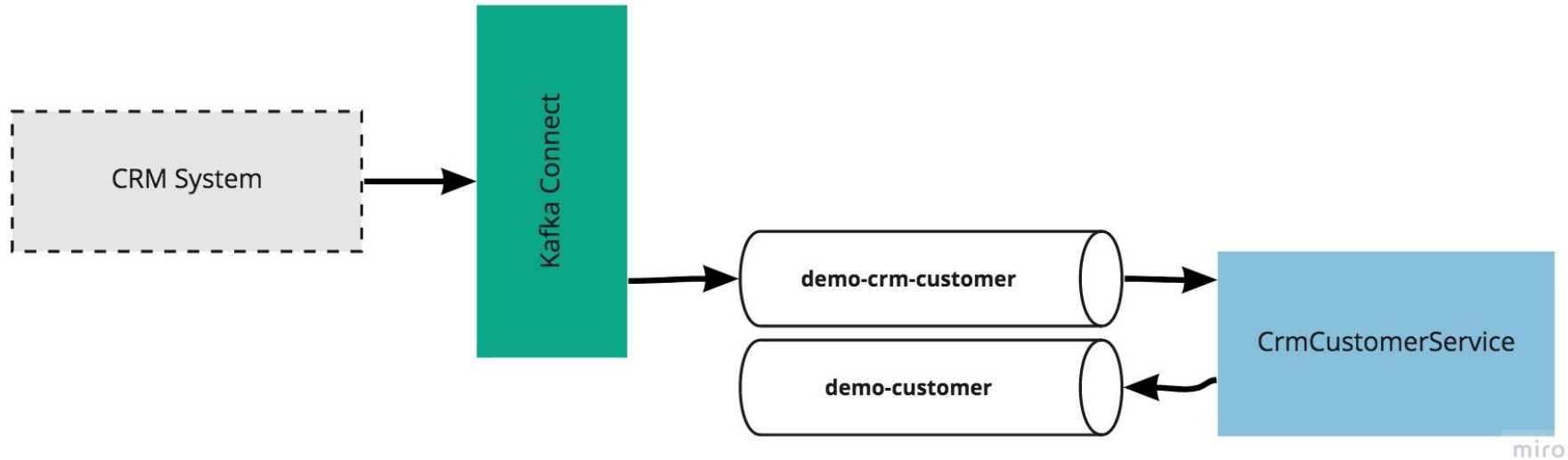
System in production



Demo system



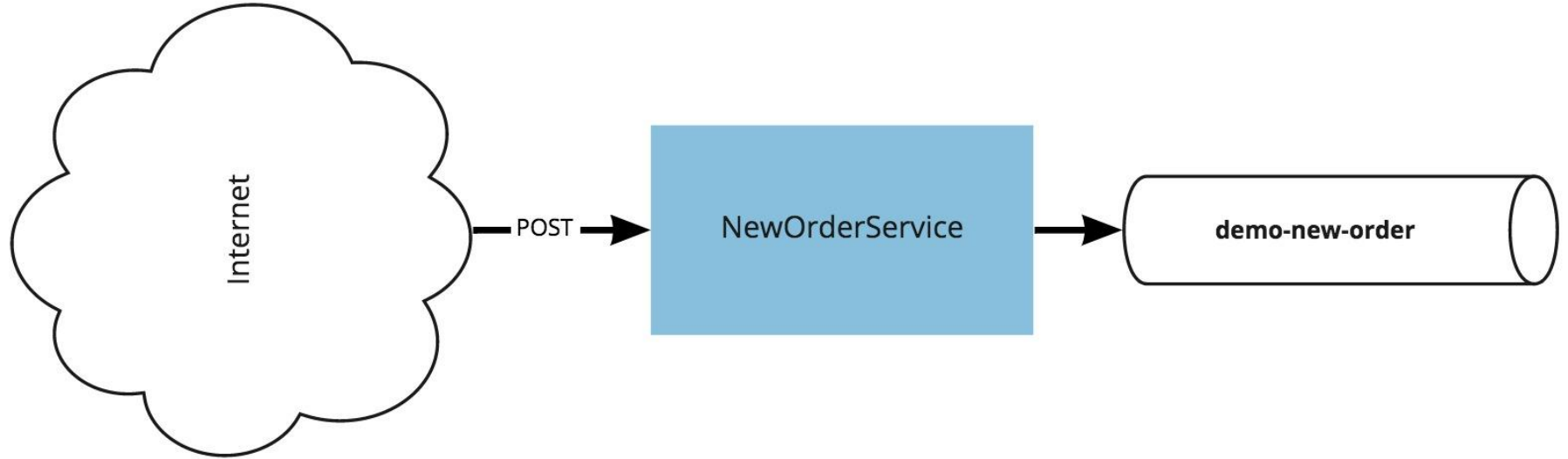
CrmCustomerService



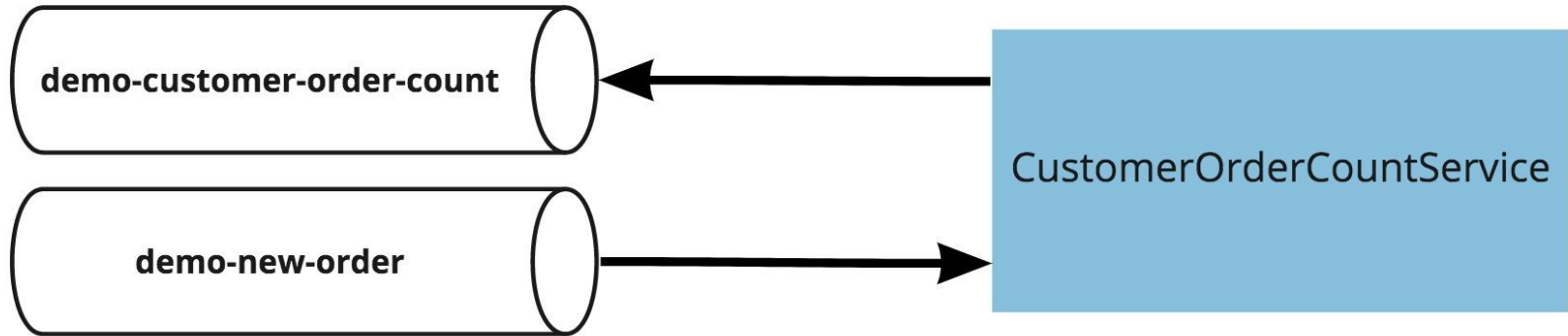
CrmCustomerService

- filter
- map
- flatMap
- selectKey

NewOrderSystem



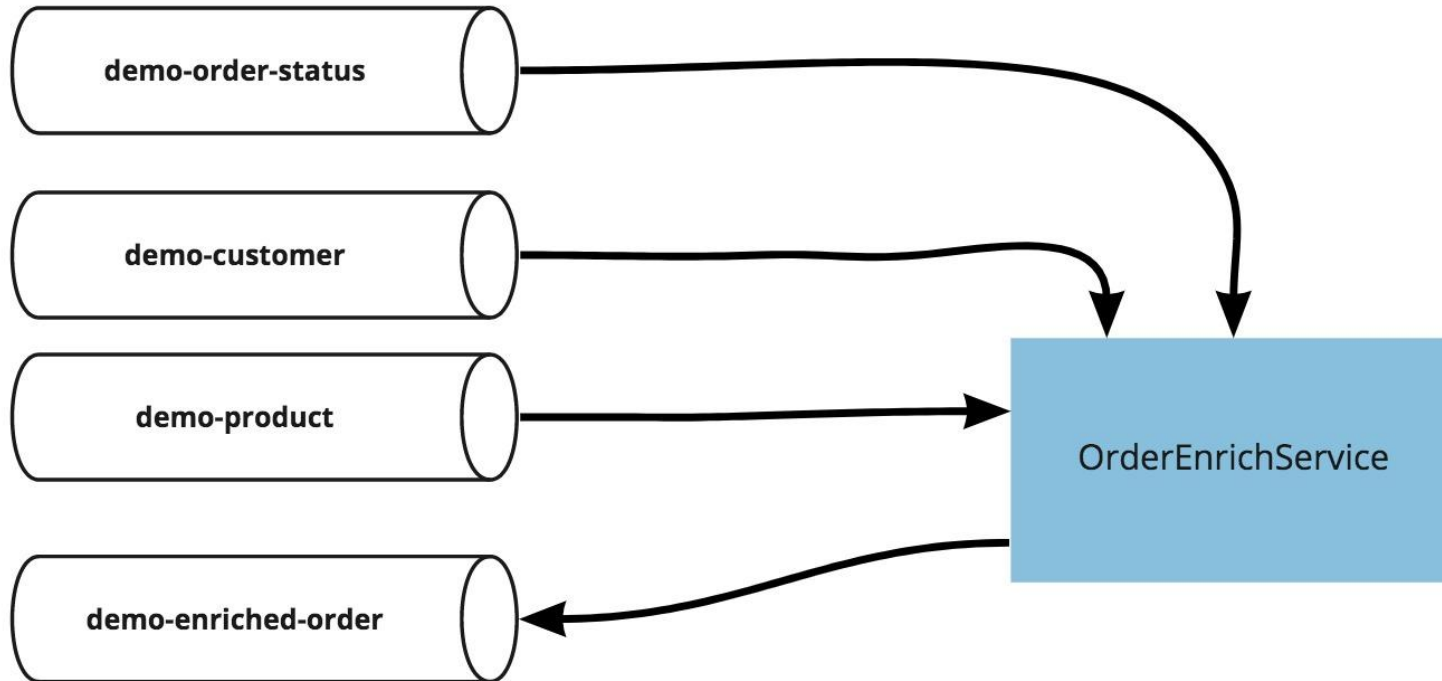
CustomerOrderCountService



CustomerOrderCountService

- groupBy
- aggregation

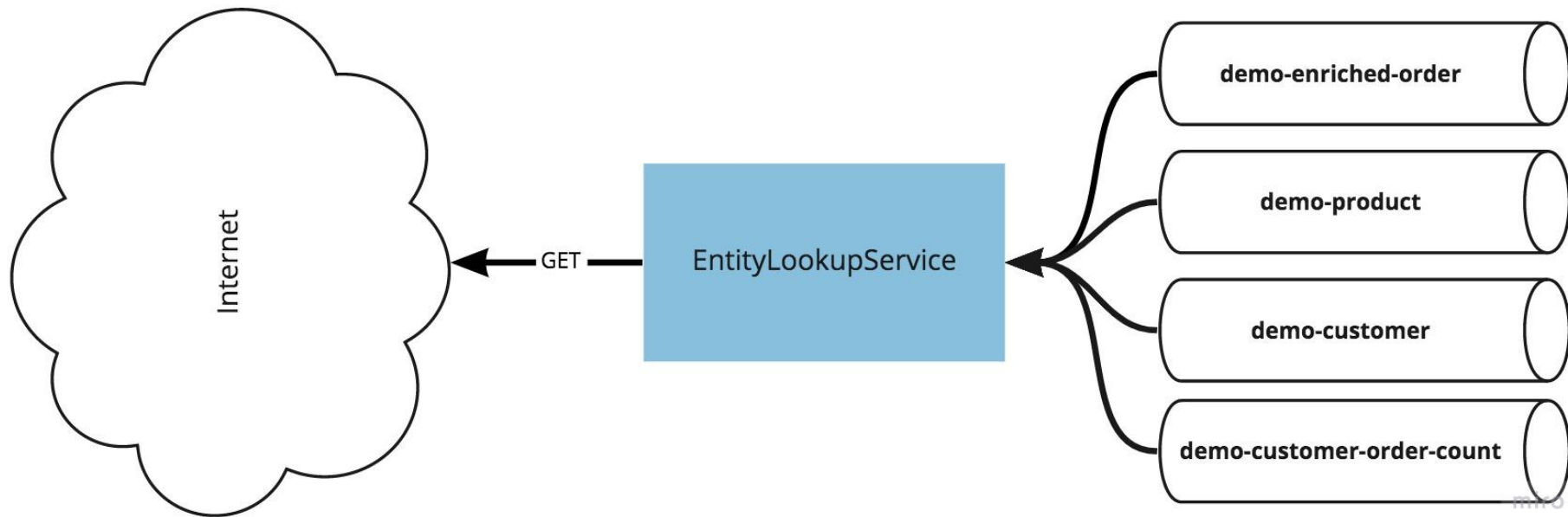
OrderEnrichService



OrderEnrichService

- stream as materialized table
- stream to global table join
- table to table join
- use stream as global state for lookup

EntityLookupService



EntityLookupService

- stream to materialized table
- no stream output
- querying state

Questions?
