

# Querying Storm Data Streams Using Trident

---



**Swetha Kolalapudi**

CO-FOUNDER, LOONYCORN

[www.loonycorn.com](http://www.loonycorn.com)

# Overview

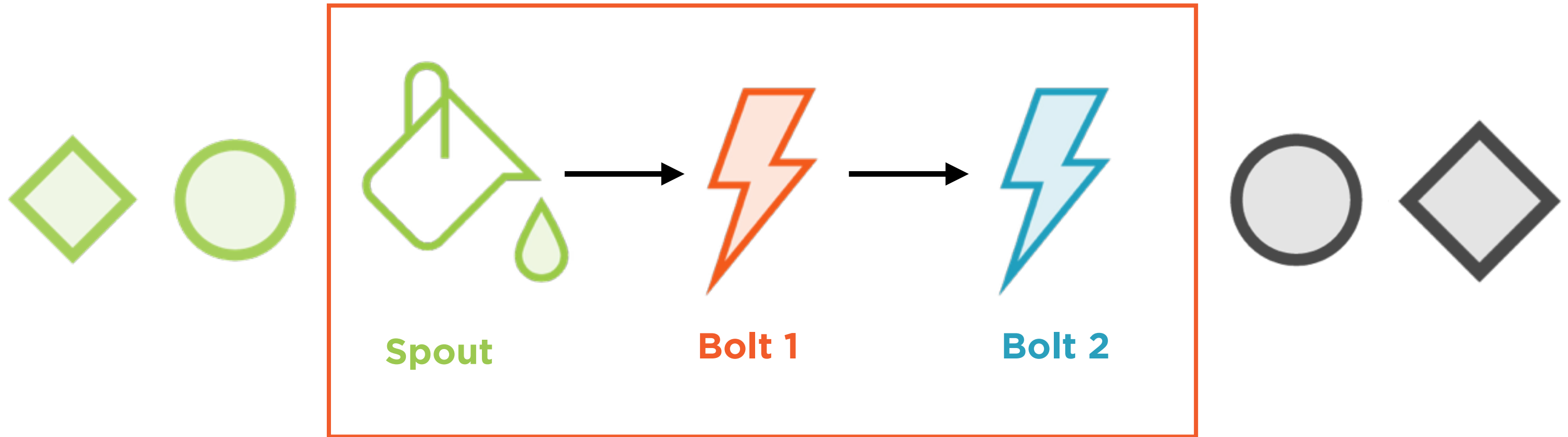
**Process data streams using the Trident API**

**Use Trident for Distributed Remote Procedure Calls i.e. DRPC**

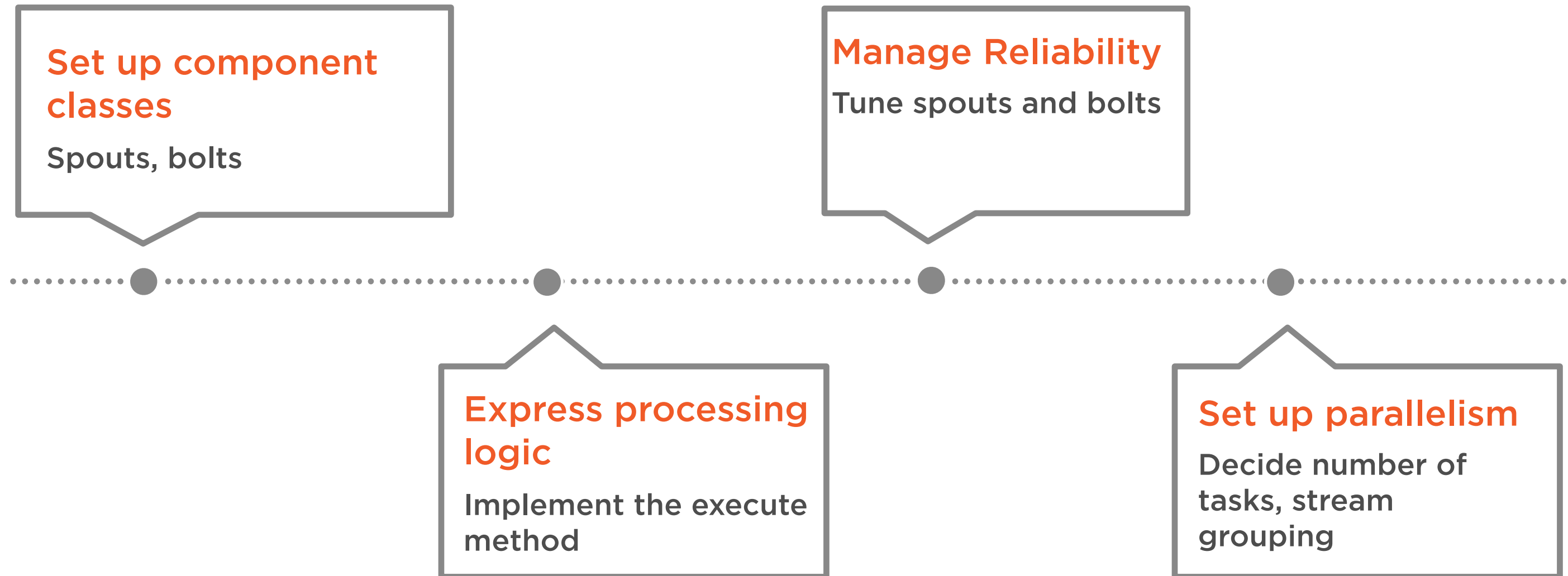
**Maintain the state of a data stream**

**Query a state object**

# Storm Topology



# Setting Up a Storm Topology



# Setting Up a Storm Topology

**Set up component classes**

Spouts, bolts



**Manage Reliability**

Tune spouts and bolts



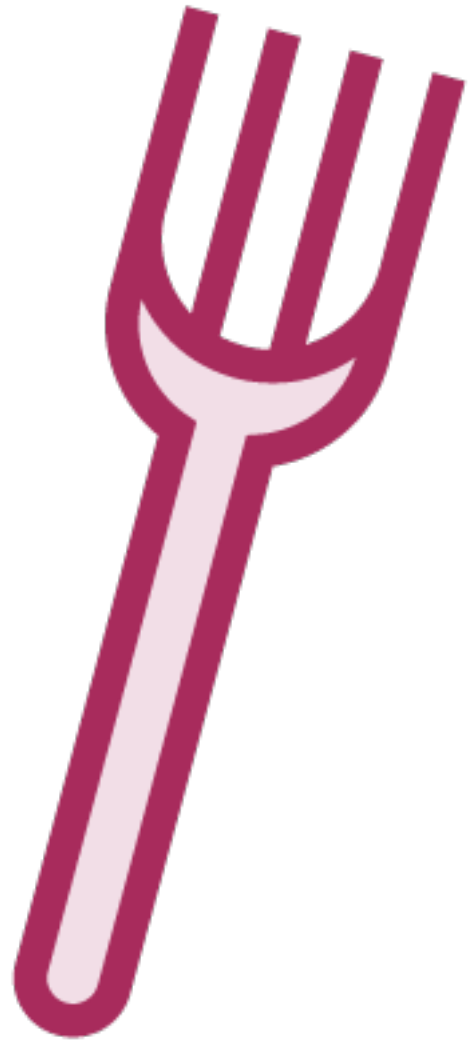
**Express processing logic**

Implement the execute method

**Set up parallelism**

Decide number of tasks, stream grouping



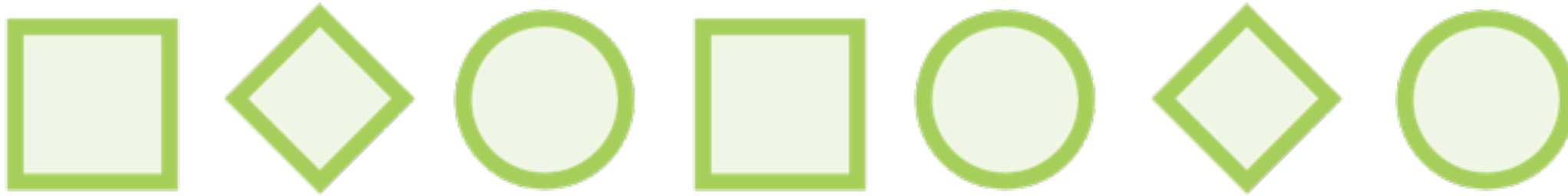


# Trident API

**Implement processing logic  
using functions**

**Apply a series of functions  
on the input stream**

# Trident Topology



## A Trident Stream object

- Generated by a spout
- Result of an operation on another Stream

# Trident Topology

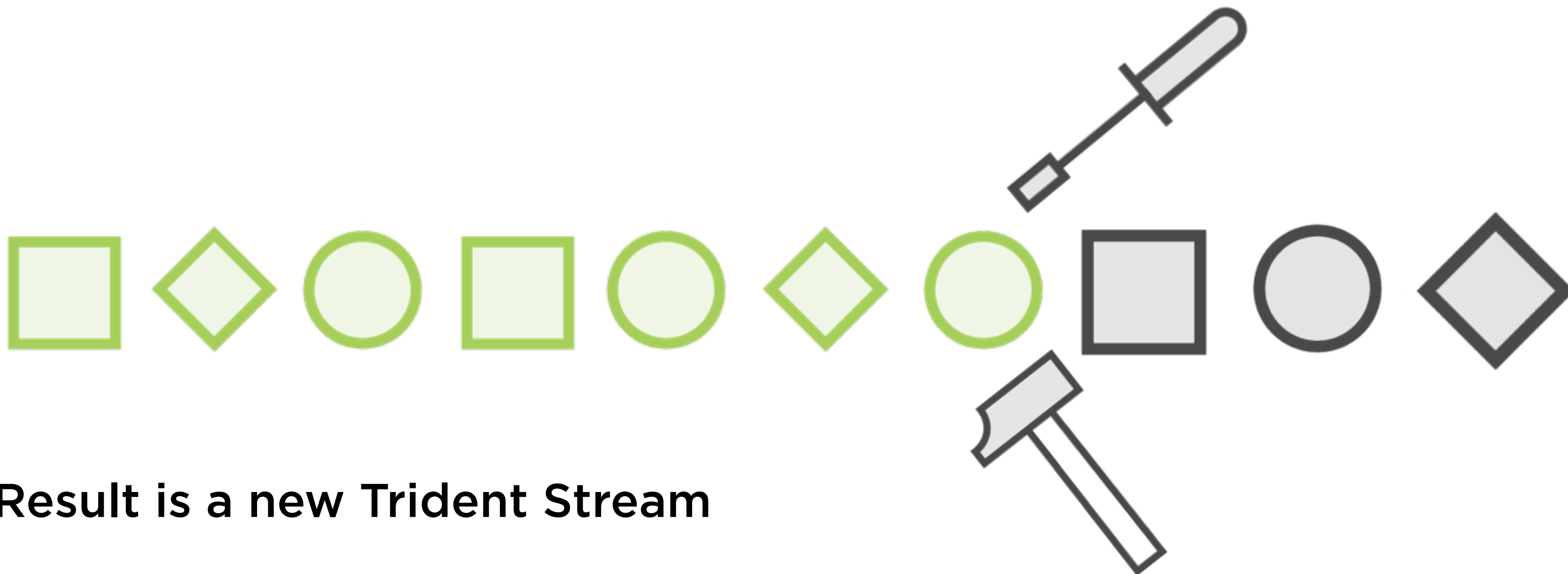


## Apply a Trident operation

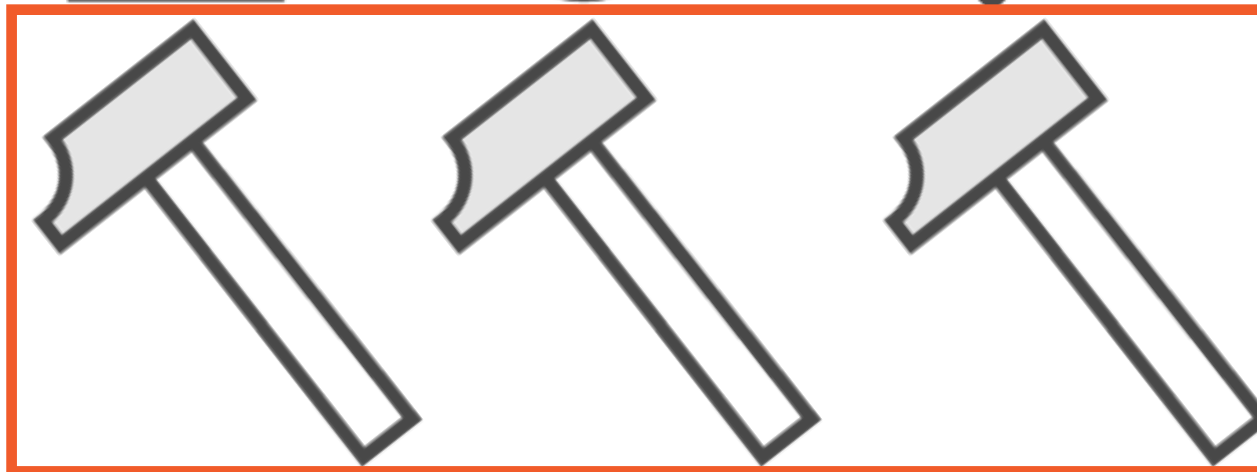
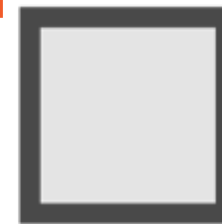
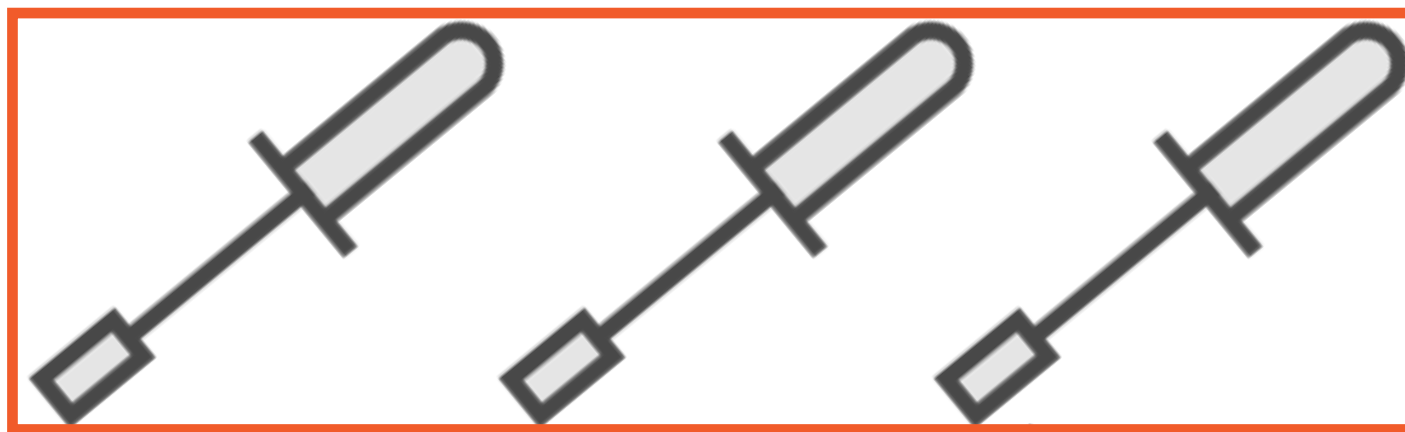
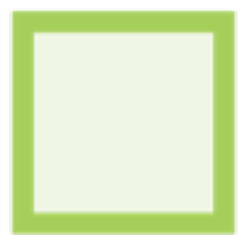
- each
- filter
- aggregate



# Trident Topology



# Trident Topology





# Trident API

**Abstract away details like  
parallelism, reliability**

**Think functionally**

# A Hello World Trident Topology

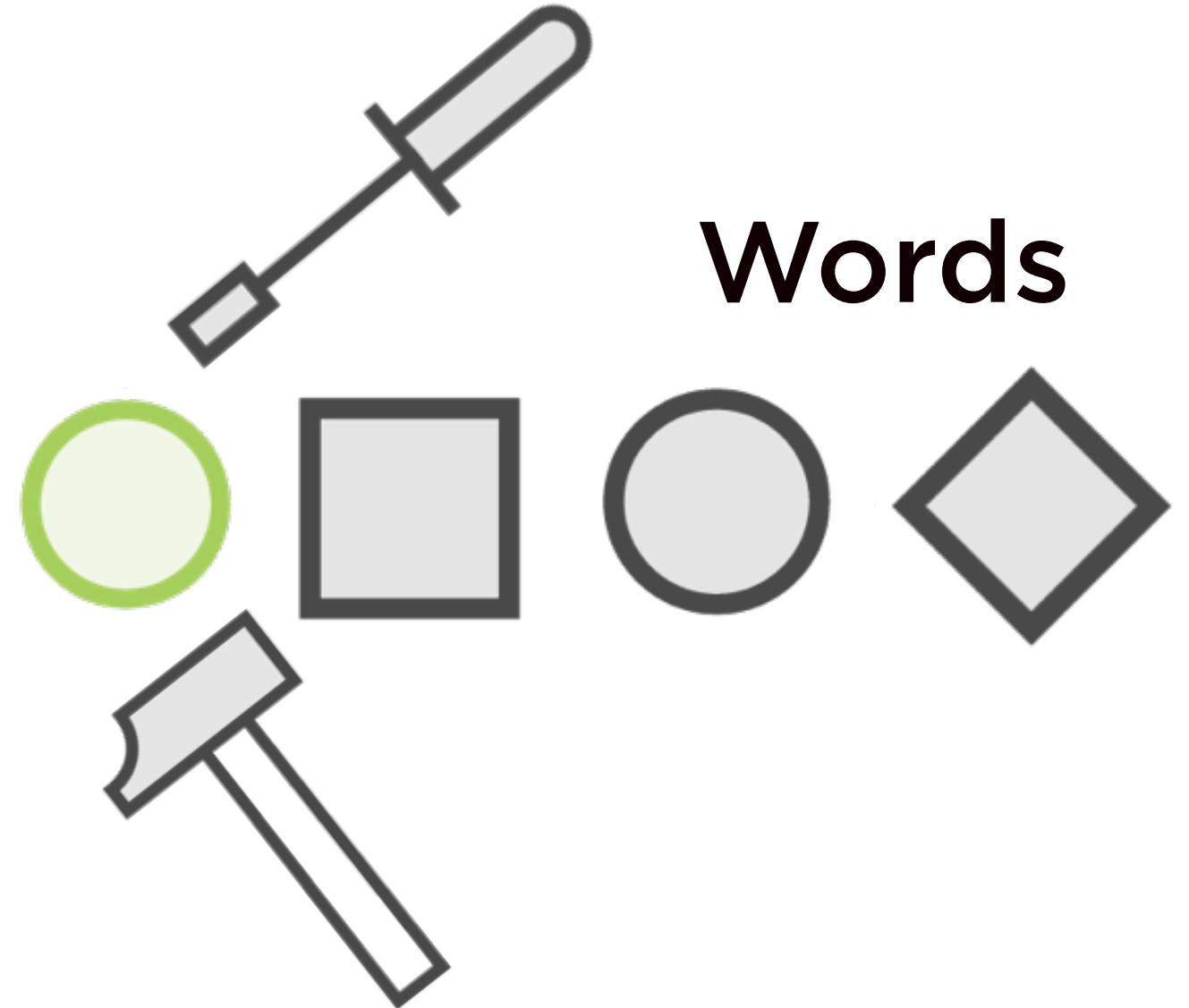
---

# Trident Hello World

**Phrases**



**Words**



# Trident Hello World

## Set up a Spout class

Use the WordReader  
spout

## Set up the topology

## Express processing logic

Extend the  
BaseFunction class

# Demo

**Set up a hello world Trident topology**

**Run the topology**

- Local
- Remote

```
public void execute(TridentTuple tuple, TridentCollector collector) {}
```

---

Extend the BaseFunction Class

**Process the input tuple**



```
public void execute(TridentTuple tuple, TridentCollector collector) {}
```

---

Extend the BaseFunction Class

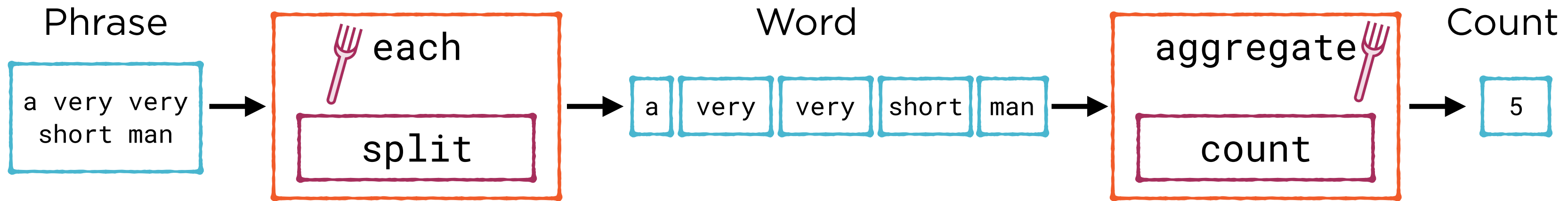
**Emit the output tuple(s)**

# Demo

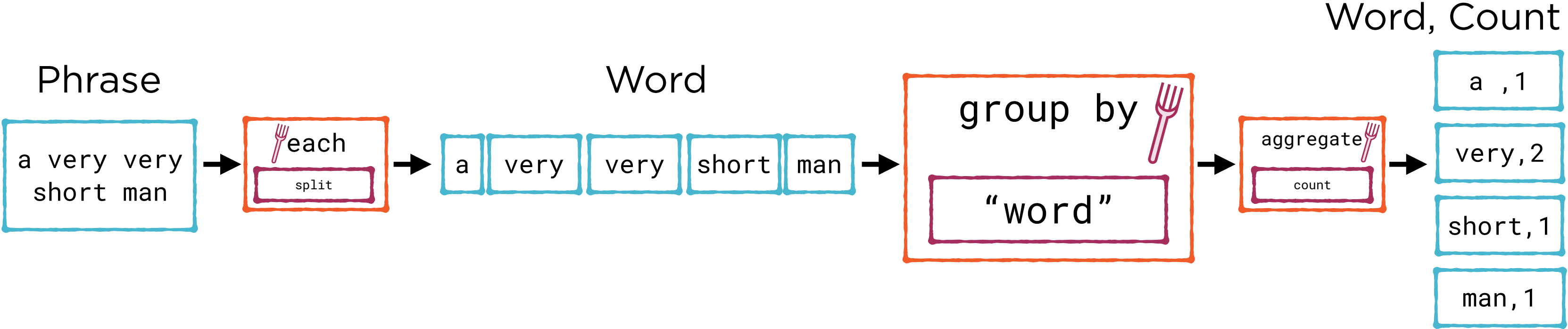
## **Apply aggregations on data streams**

- aggregate
- group by

# Trident Aggregation



# Trident Aggregation



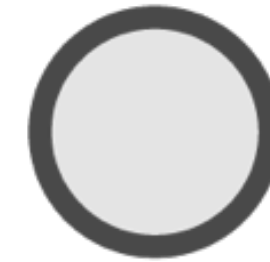
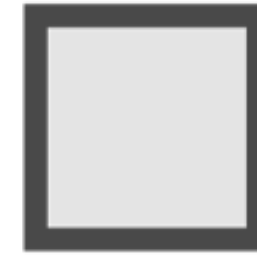
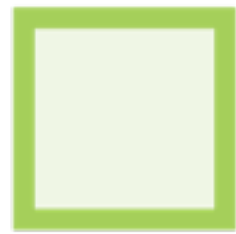
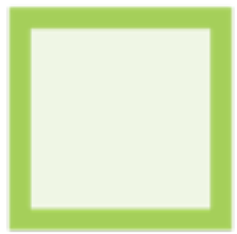
# Distributed Remote Procedure Calls Using Trident

---

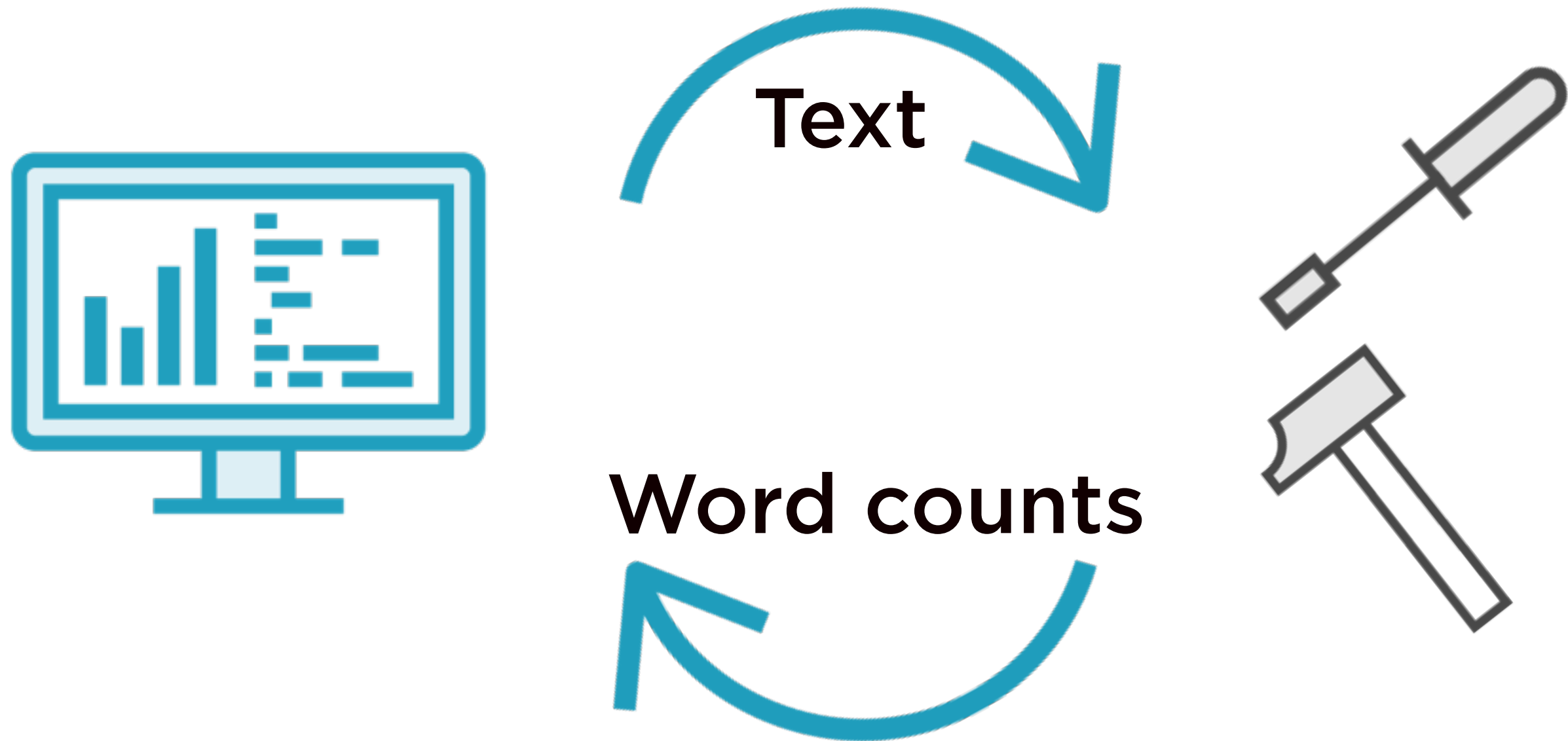
# Real-time Application

**Text**

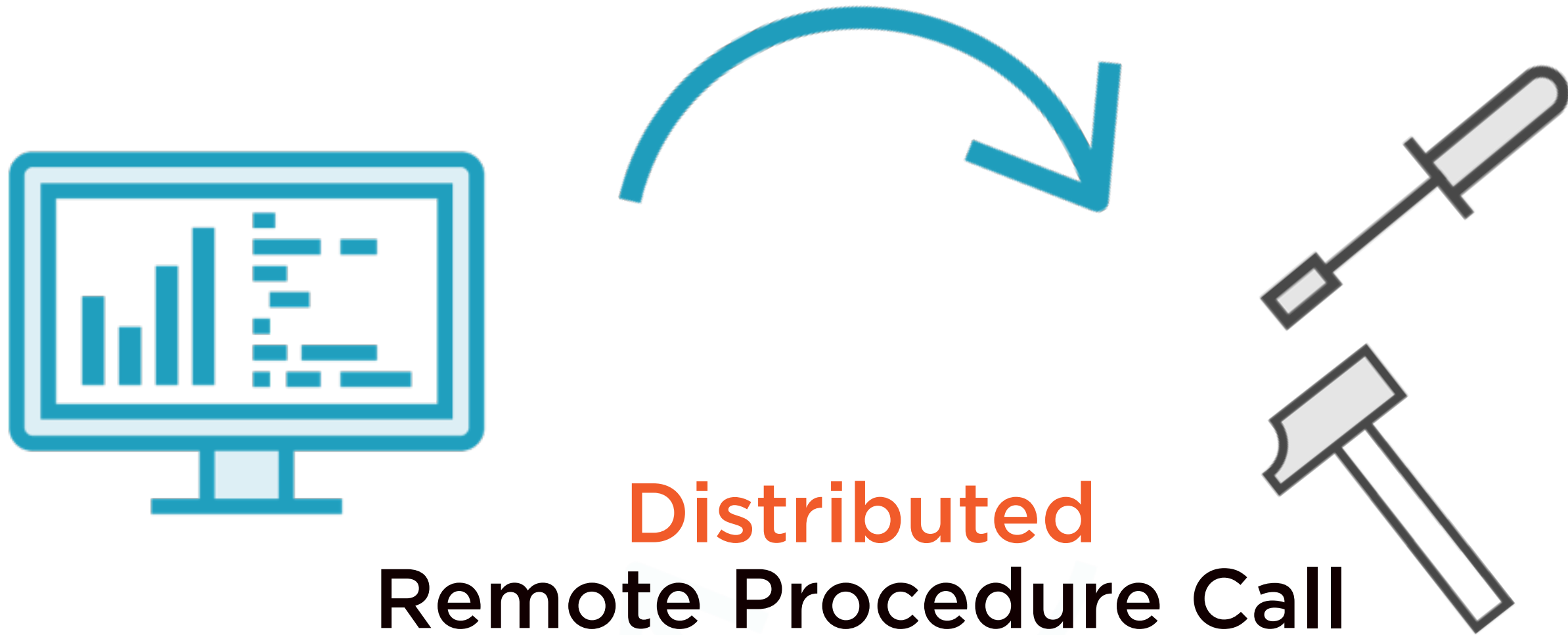
**Word counts**



# Real-time Application



# Real-time Application

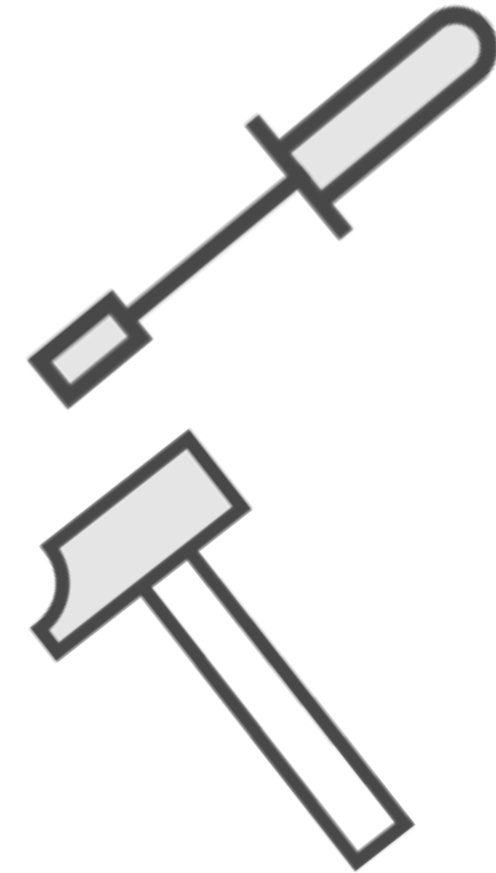




# Real-time Application



**DRPC Client**

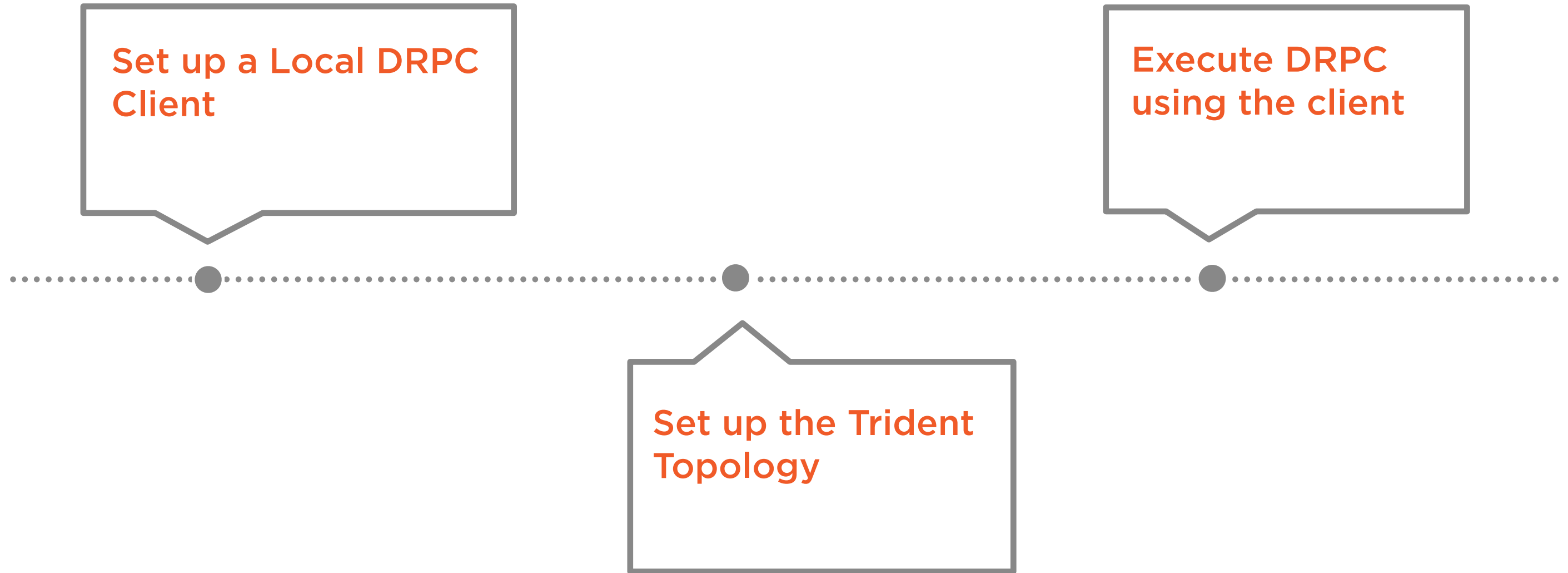


# Trident DRPC Topology

Set up a Local DRPC Client

Execute DRPC using the client

Set up the Trident Topology



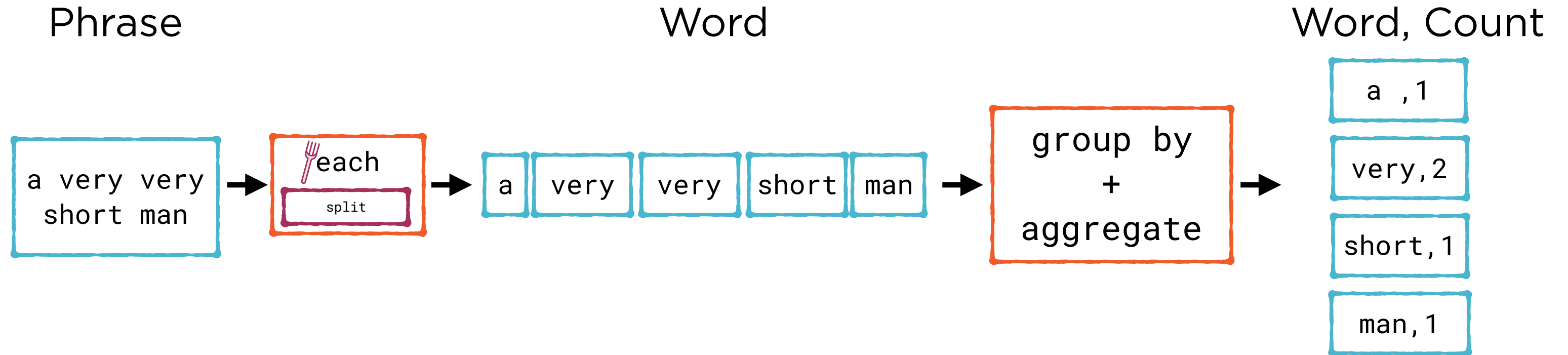
Demo

**Set up a DRPC topology**

# Maintaining and Querying States

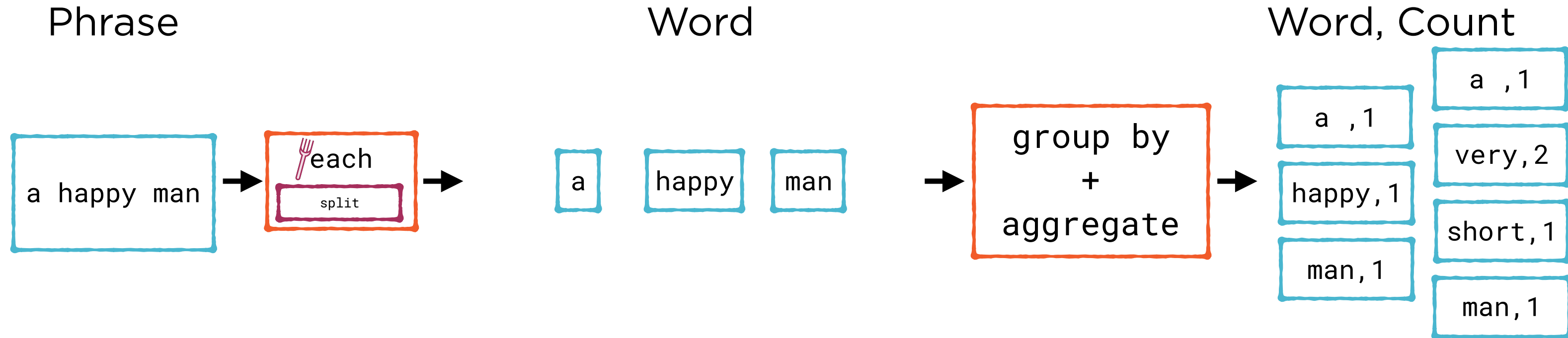
---

# Word Count Topology



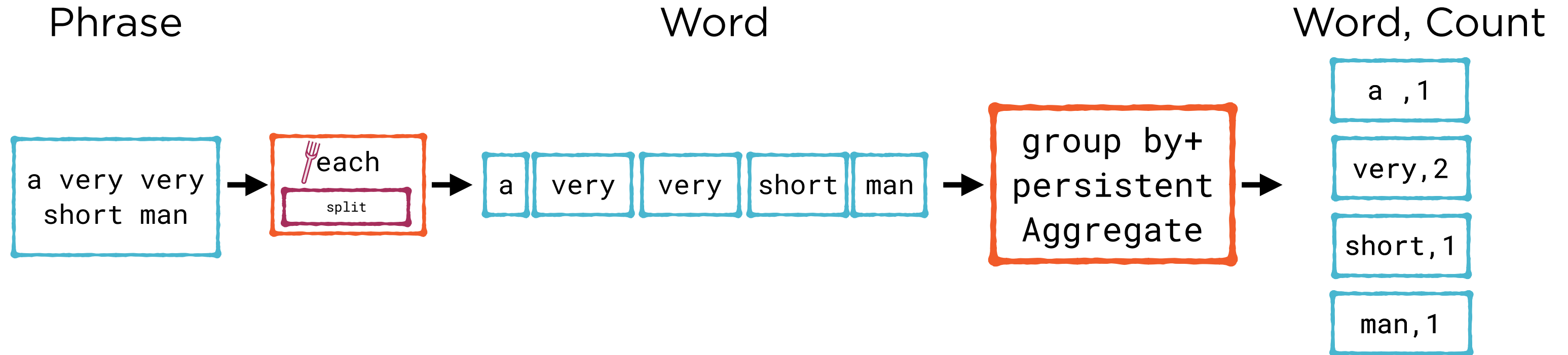
**Result of aggregate is  
a Stream object**

# Word Count Topology



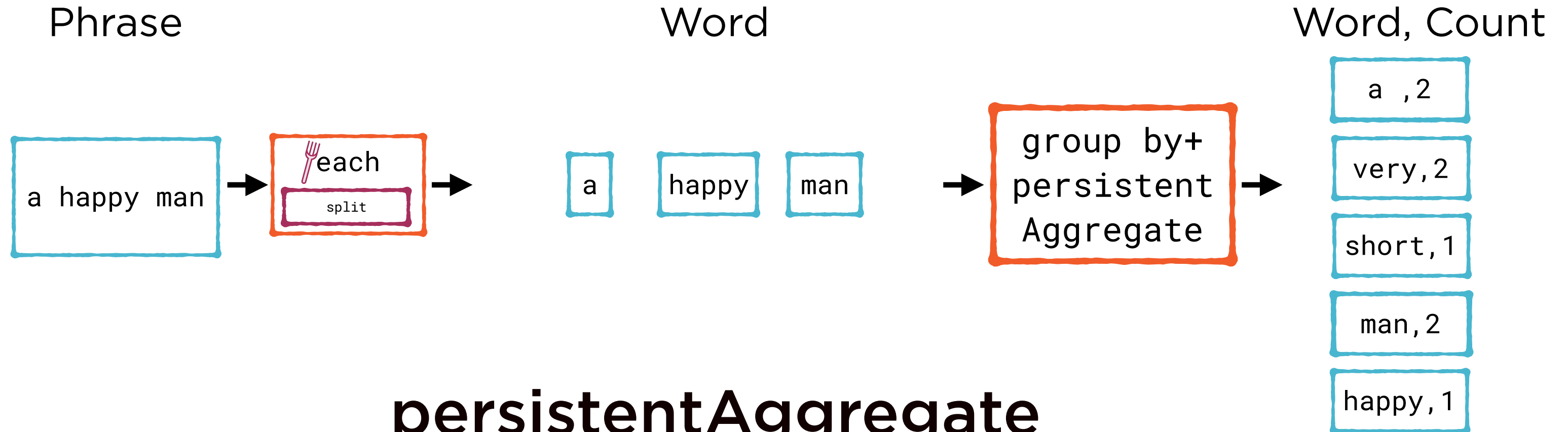
**Result of aggregate is  
a Stream object**

# Word Count Topology



**persistentAggregate**  
stores results in a  
**State object**

# Word Count Topology



**persistentAggregate**  
stores results in a  
**State object**



Word, Count

a , 2

very, 2

short, 1

man, 2

happy, 1

## Trident State

**May be in-memory or persistent**

**Can be queried for information**

# Demo

**Set up a Trident State object to store word counts**

**Query the state object for counts of specific words**

# Summary

**Process data streams using the Trident API**

**Use Trident for Distributed Remote Procedure Calls i.e. DRPC**

**Maintain the state of a data stream**

**Query a state object**