Applying Real-time Processing Using Apache Storm

UNDERSTANDING THE COMPONENTS OF STORM

Swetha Kolalapudi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Differentiate between real-time processing and batch processing

Understand the components of a Storm topology

Understand how data is represented

Implement a stock price tracker topology

Quant Trading

Quant traders at a hedge fund

Evaluate stock performance

Performance Reports

Fixed frequency

- Quarterly, Monthly, Weekly, Daily etc

Compute metrics over the period

- High, low, average, momentum etc

Build models from historical data

Batch Processing

Monitoring, Alerts, Actions

Monitor the stock price in real-time

Apply trading models to identify actions

Alerts/triggers for specific trades

Real-time Processing

Real-time Processing vs. Batch Processing

Process each event as it happens	**Process a group of events together**
Continuous processing	**On-demand/At fixed frequency**
Requires low latency	**High latency is acceptable**

Real-time Processing vs. Batch Processing

Storm
Heron

Spark
Flink

Hadoop
MapReduce

Apache Storm

Open-sourced originally by Twitter

Real-time processing counterpart for Hadoop

Analytics, transformations, machine learning in real-time

Low latency due to distributed architecture

Components of a Storm Topology

Storm Application

Data is received as a stream

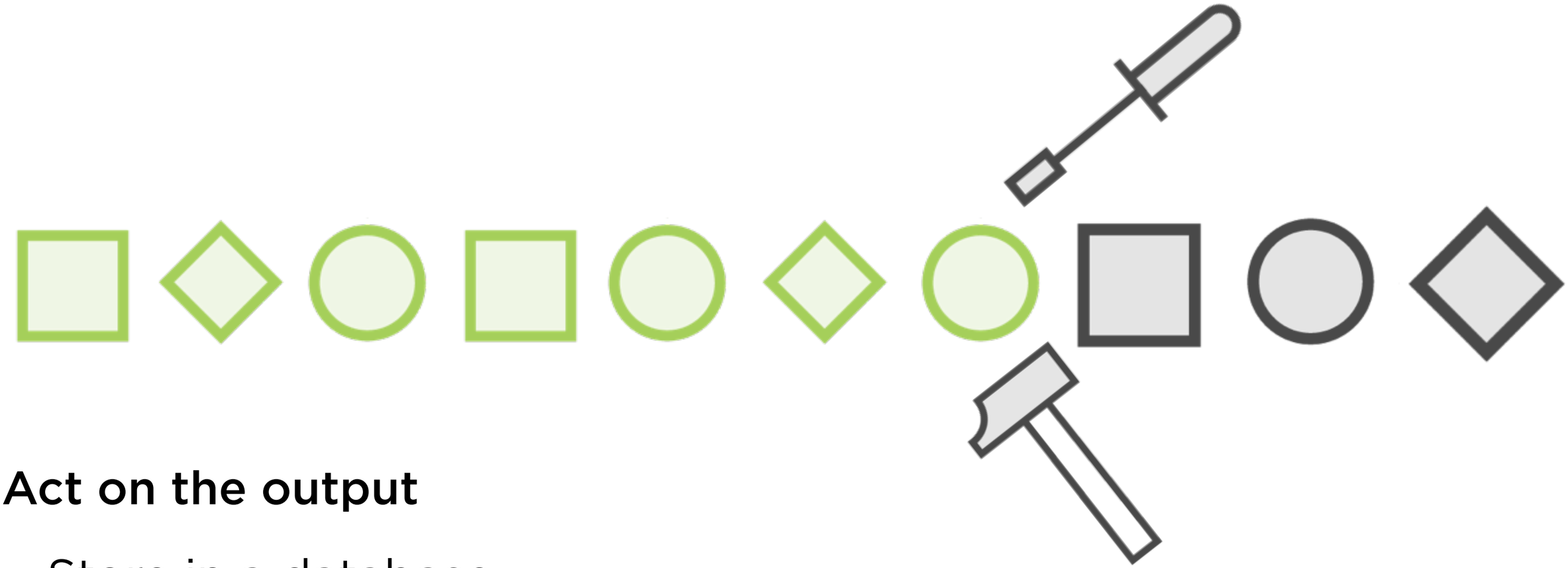
- Stock prices
- Log messages
- Tweets

Storm Application

Process the data one entity at a time

- Compare the price to a constant
- Monitor error messages
- Find latest trending hashtags

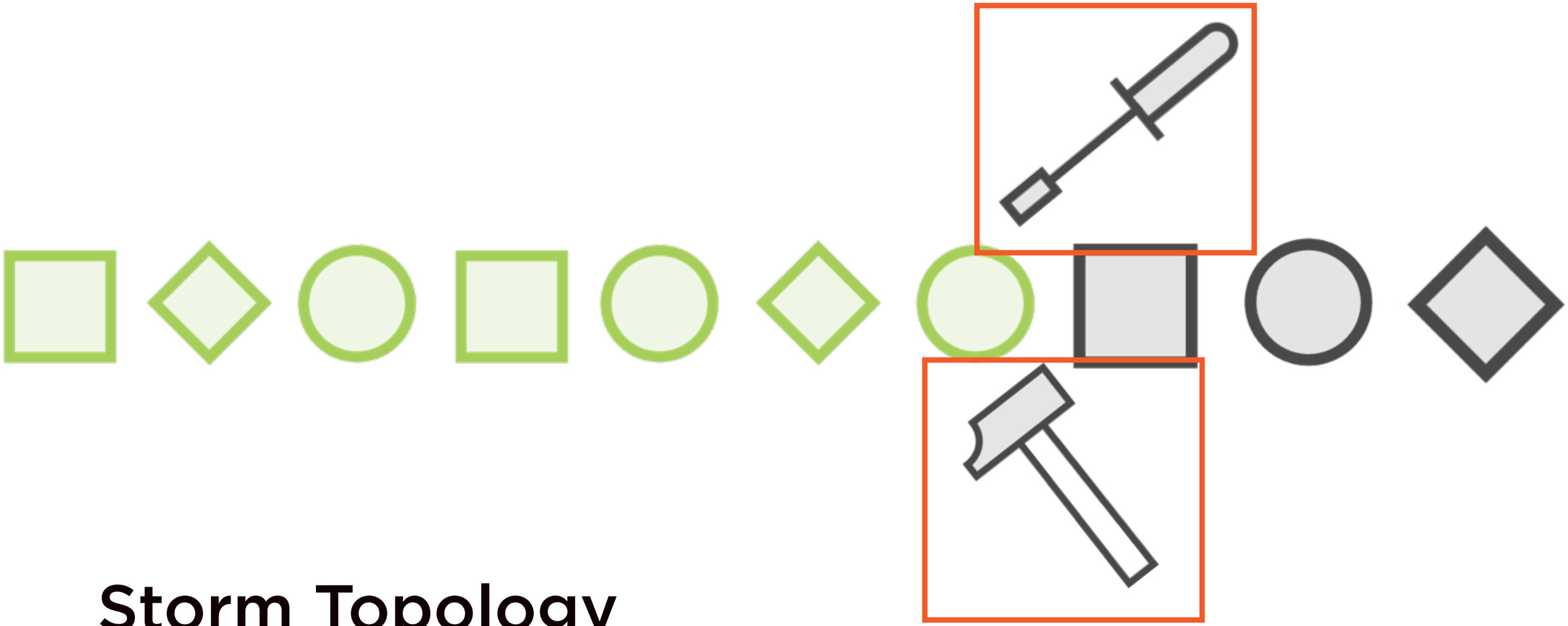
Storm Application



Act on the output

- Store in a database
- Trigger an alert

Storm Application



Storm Topology

Components of a Storm Topology

Spout

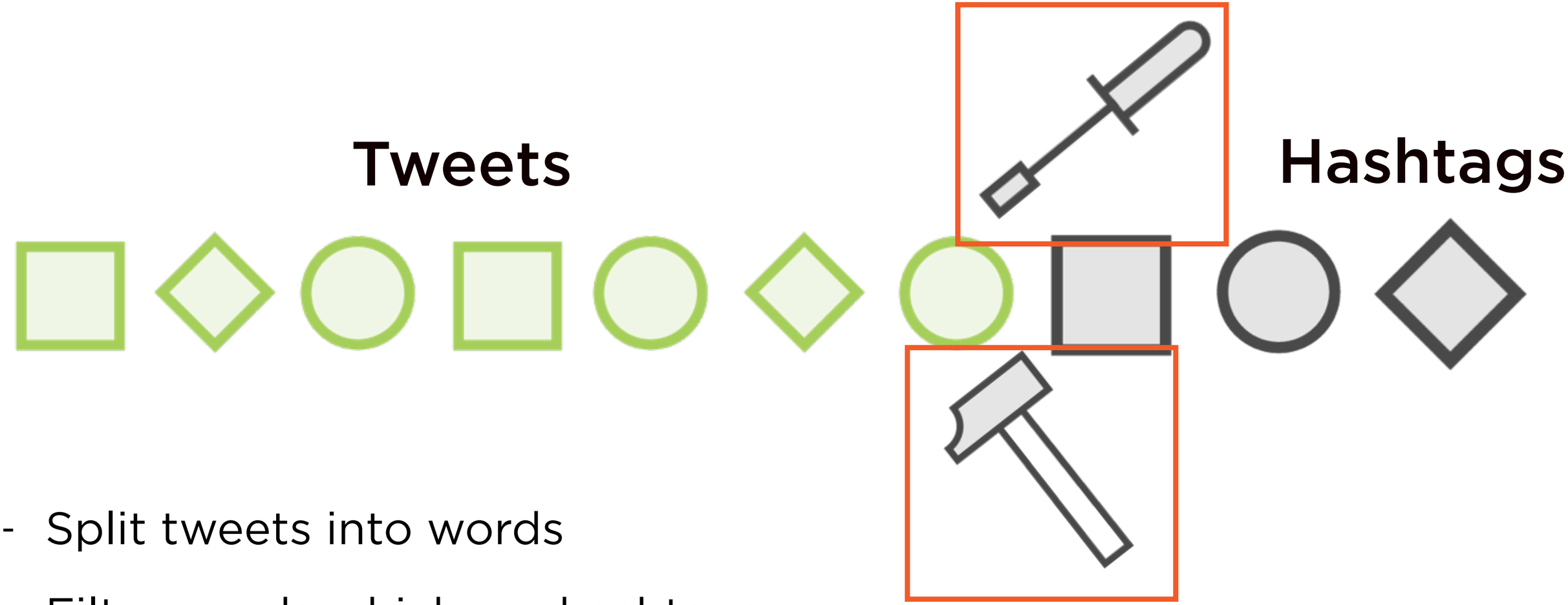
Receives data from the source

Emits it to the rest of the topology

Bolt

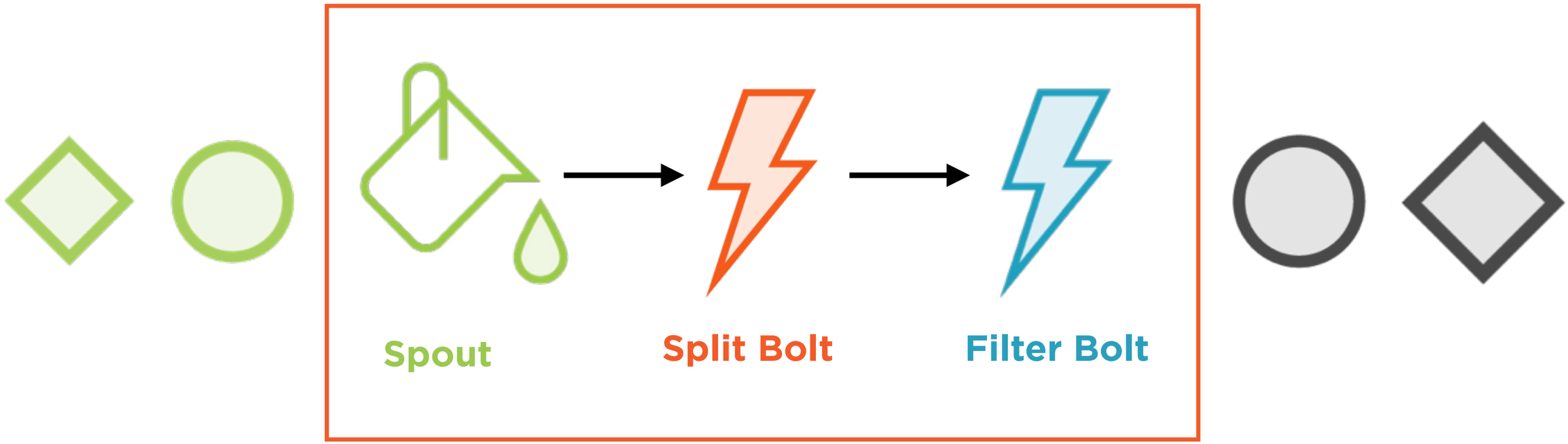
Processes the data

Hashtag Extractor

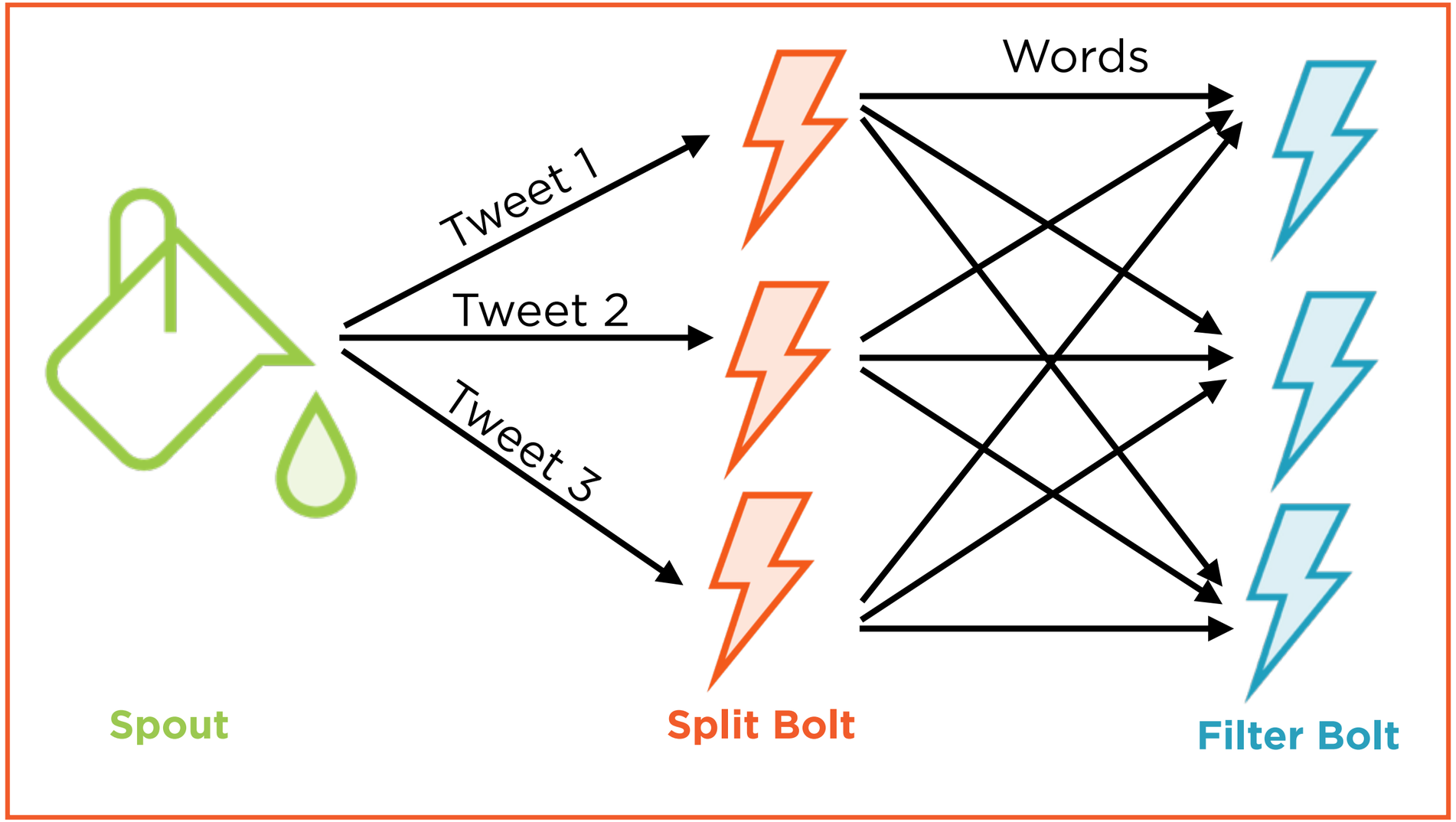


- Split tweets into words
- Filter words which are hashtags

Hashtag Extractor Topology



Hashtag Extractor Topology

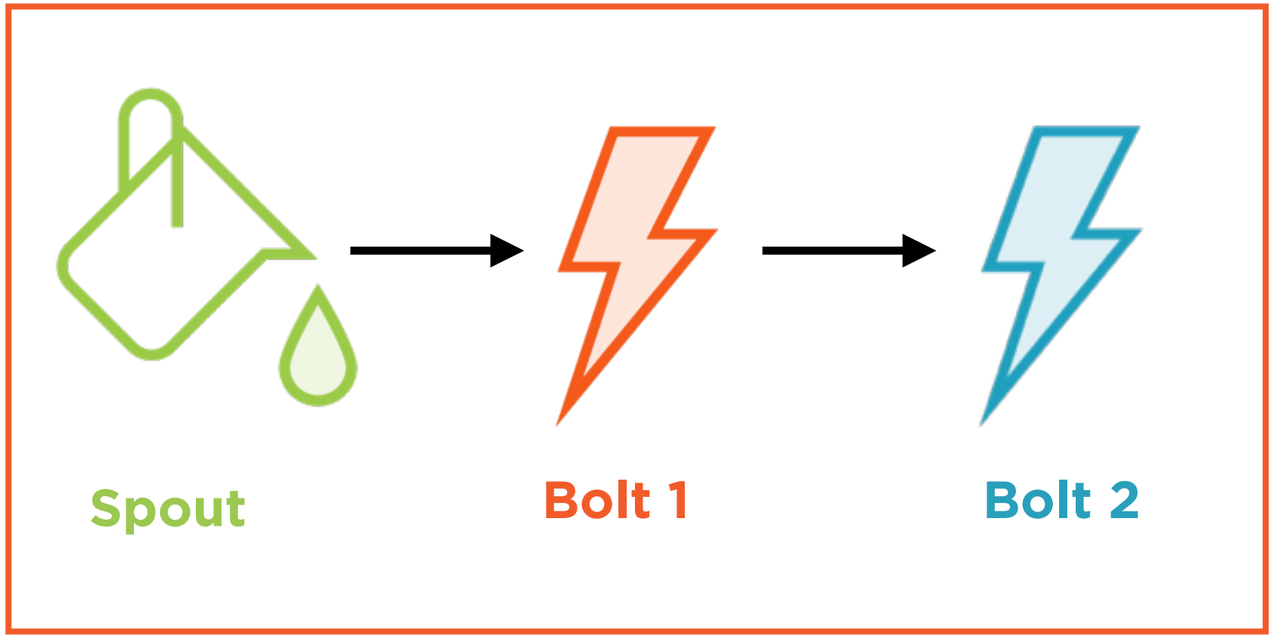


Components run by Java processes

Need a central coordinating process

All processes run in a storm cluster

Storm Topology



Storm Cluster

Local

All processes run locally

Run topologies from IDE

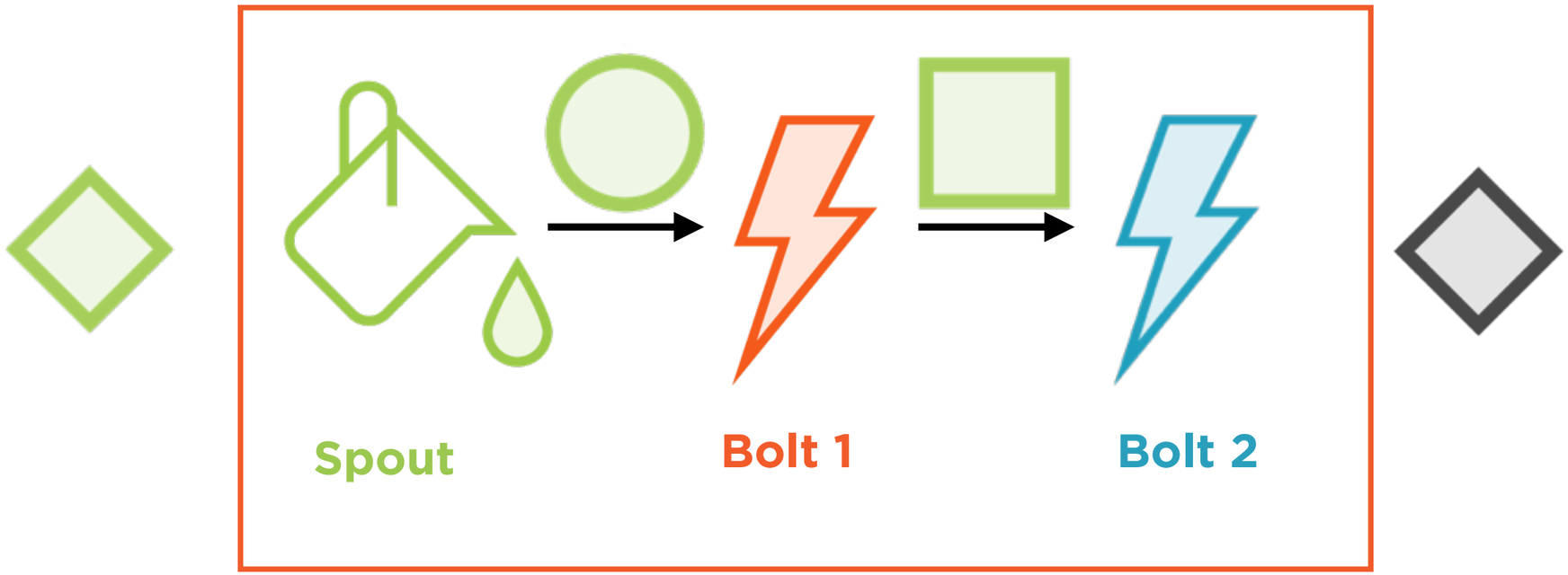
Remote

All processes run on different machines

Submit a jar to cluster

Representing Data in Storm Topologies

Storm Topology



Components emit Tuples

Storm Tuples

Atomic data points

Named tuples

Symbol	Datetime	Price
MSFT	Mar 1, 12 9:00 AM	63

Storm Tuples

Fields

Schema of the tuple

Symbol	Datetime	Price
MSFT	Mar 1, 12 9:00 AM	63

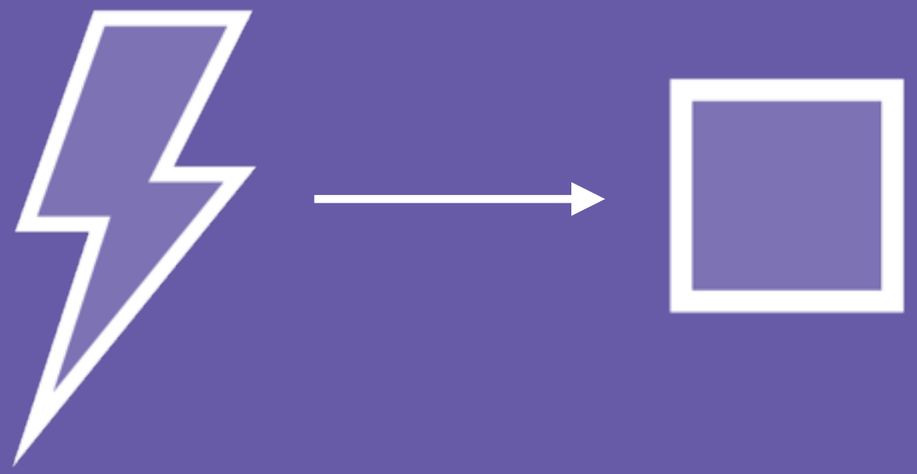
Storm Tuples

Values

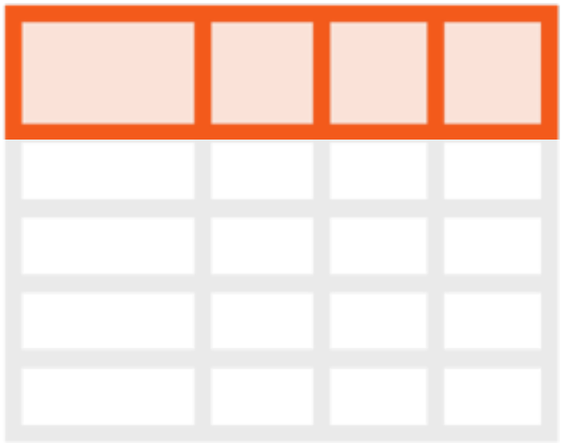
Data in the tuple

Symbol	Datetime	Price
MSFT	Mar 1, 12 9:00 AM	63

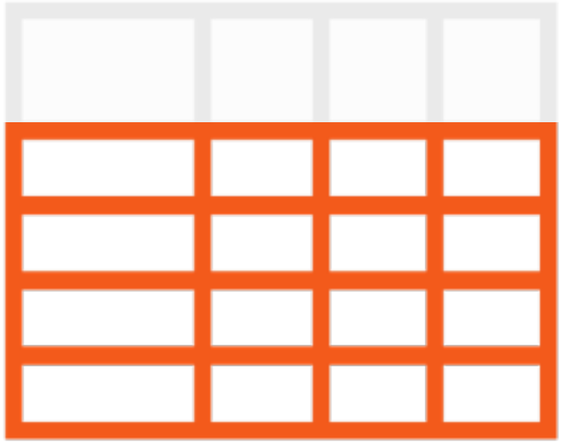
Each storm component emits
Tuples with a **fixed schema**



Setting up a Storm Component

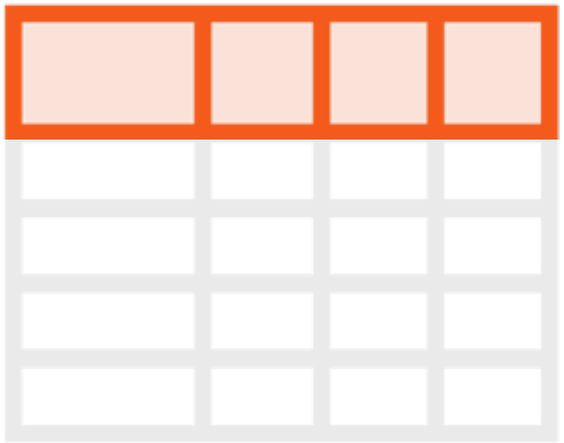


Declare Fields



Emit Values

Setting up a Storm Component



Declare Fields

Emit Values

```
public void declareOutputFields(OutputFieldsDeclarer declarer){
declarer.declare(new Fields("symbol","date","price"));}
```

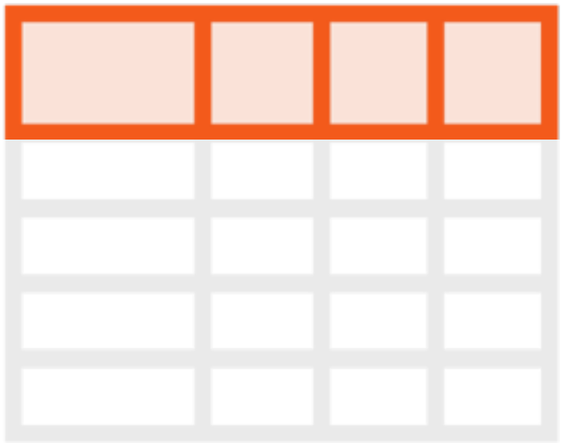
Declare Fields

Implement this method in every storm component

```
public void declareOutputFields(OutputFieldsDeclarer declarer){
declarer.declare(new Fields("symbol","date","price"));}
```

Declare Fields

Setting up a Storm Component

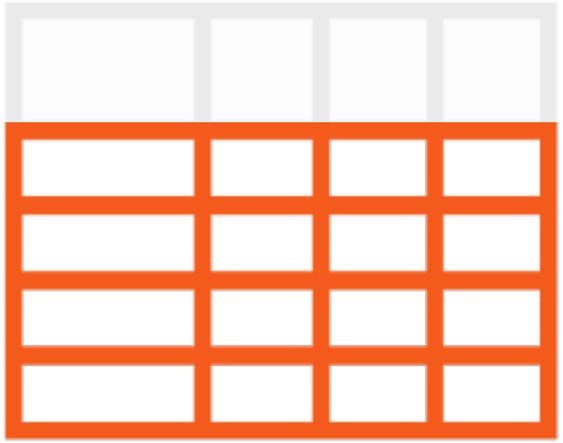


Declare Fields

Emit Values

Setting up a Storm Component

Declare Fields



Emit Values

```
collector.emit(new Values("MSFT","Mar 12 9:00 AM",62.5))
```

Emit Values

All components use a collector object to emit values

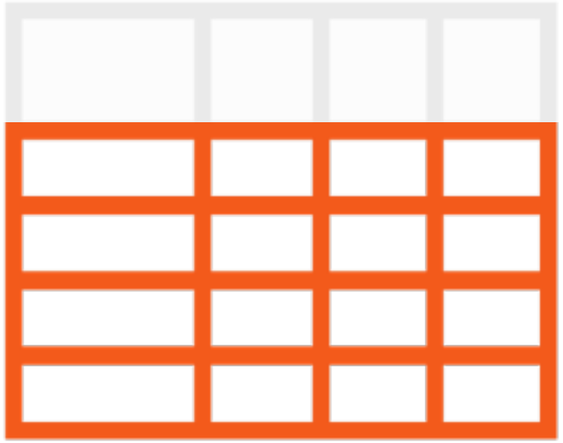
```
collector.emit(new Values("MSFT","Mar 12 9:00 AM",62.5))
```

Emit Values

Provide Objects in the same order as declared Fields

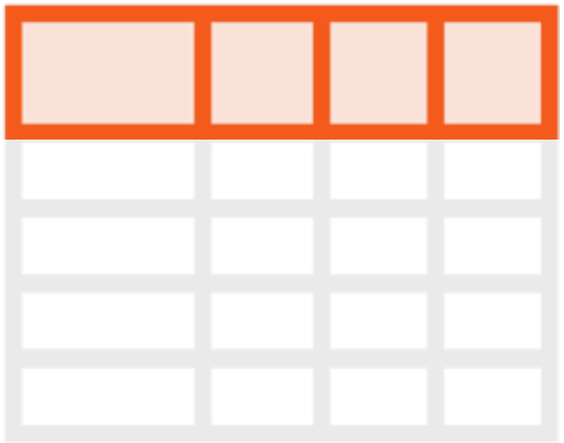
Setting up a Storm Component

Declare Fields

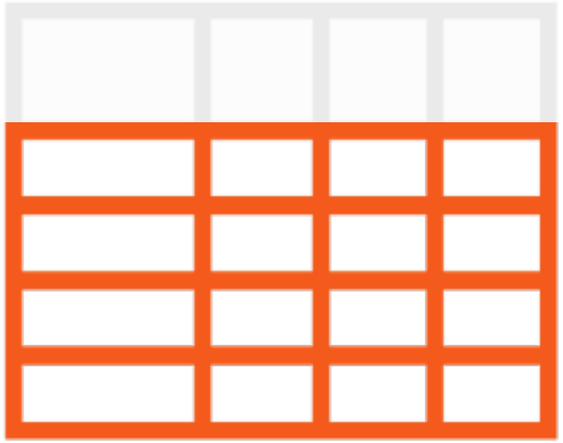


Emit Values

Setting up a Storm Component



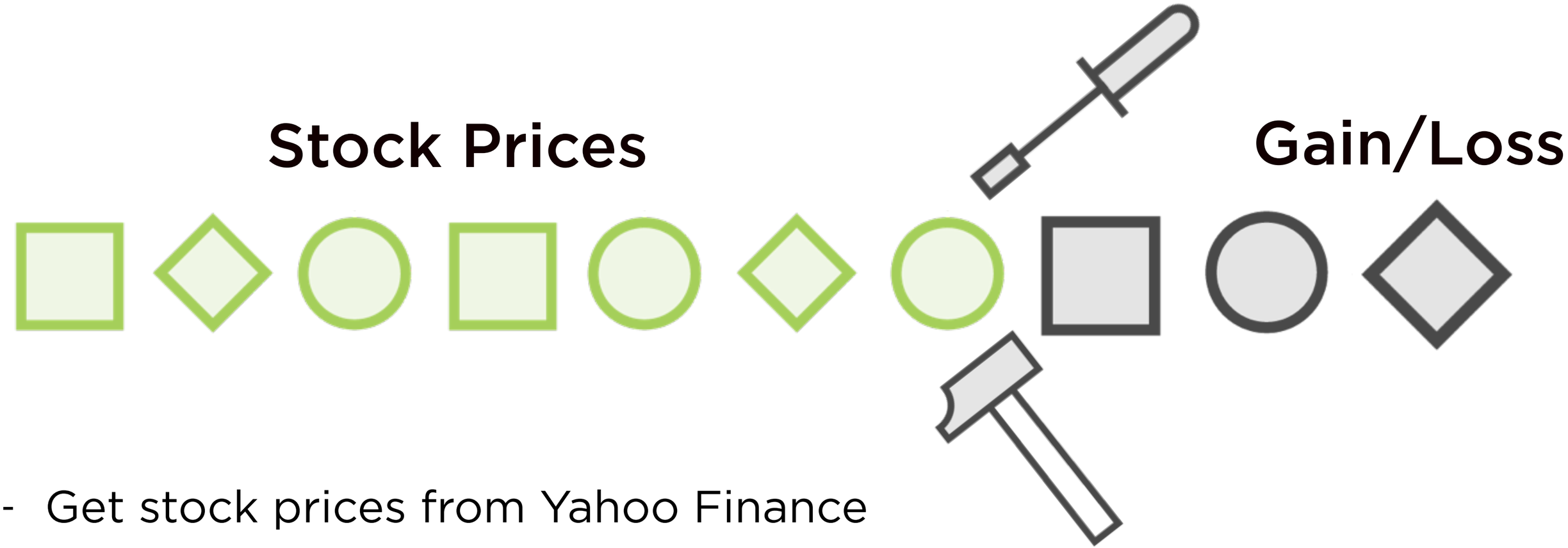
Declare Fields



Emit Values

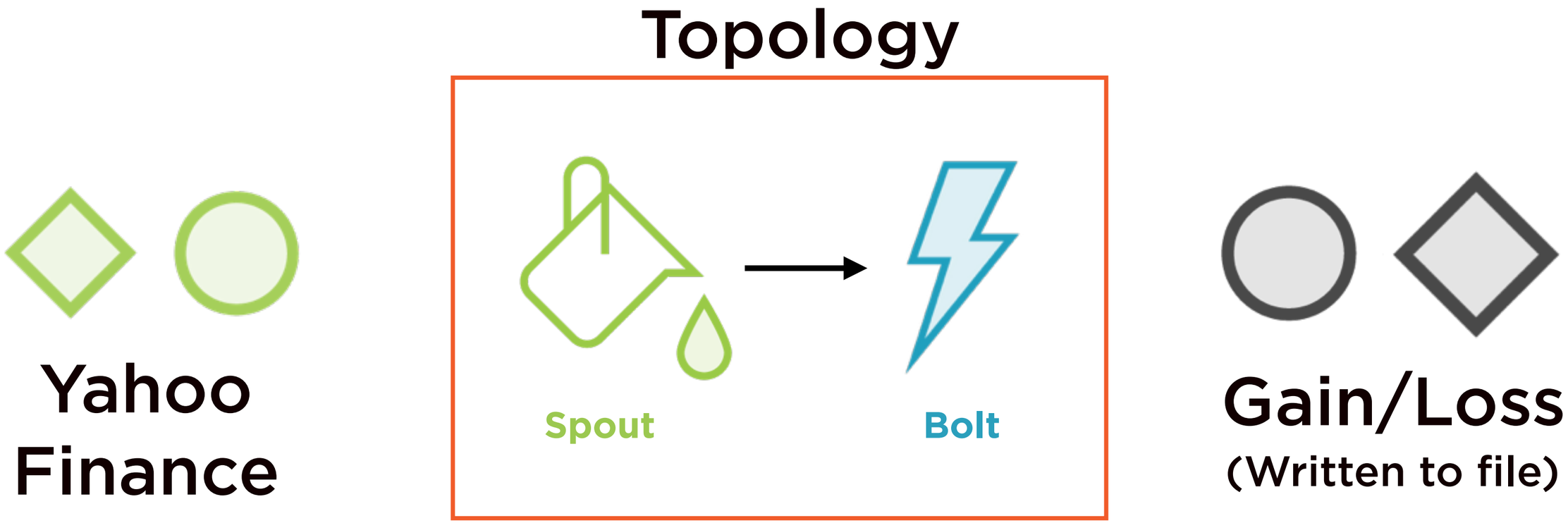
Building a Stock Price Tracker

Stock Price Tracker



- Get stock prices from Yahoo Finance
- If price>previous close, **Gain** else **Loss**

Stock Price Tracker



Stock Price Tracker

Set up a project

Use Maven to add Storm as a dependency

Set up the Spout

Extends the BaseRichSpout abstract class

Set up the Bolt

Extends the BaseBasicBolt abstract class

Run the topology

Build the topology

Submit to a local cluster

Stock Price Tracker

Set up a project

Use Maven to add Storm as a dependency

Demo

Set up a project

- Add the Storm dependency

Stock Price Tracker

Set up a project

Use Maven to add Storm as a dependency

Set up the Spout

Extends the BaseRichSpout abstract class

Demo

Set up the Spout

- Extend the BaseRichSpout class
- Connect to Yahoo Finance API
- Get stock price, previous close and send to Bolt

```
public void open(Map conf, TopologyContext context,
                 SpoutOutputCollector collector) {}

public void nextTuple() {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}
```

Implementing the Spout

```
public void open(Map conf, TopologyContext context,
                 SpoutOutputCollector collector) {}

public void nextTuple() {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}
```

Initializing the Spout

```
public void open(Map conf, TopologyContext context,
                 SpoutOutputCollector collector) {}

public void nextTuple() {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}
```

Initializing the Spout

Topology configuration

```
public void open(Map conf, TopologyContext context,
                 SpoutOutputCollector collector) {}

public void nextTuple() {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}
```

Initializing the Spout

Component name, id etc

```
public void open(Map conf, TopologyContext context,
                 SpoutOutputCollector collector) {}

public void nextTuple() {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}
```

Initializing the Spout

Used to communicate between components

- Emit data, success/failure

```
public void open(Map conf, TopologyContext context,
                 SpoutOutputCollector collector) {}

public void nextTuple() {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}
```

Initializing the Spout

```
public void open(Map conf, TopologyContext context,
                 SpoutOutputCollector collector) {}

public void nextTuple() {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}
```

Declare Fields

```
public void open(Map conf, TopologyContext context,
                 SpoutOutputCollector collector) {}

public void nextTuple() {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}
```

Emit Tuples

Called continuously as long as the spout runs

Stock Price Tracker

Set up a project

Use Maven to add Storm as a dependency

Set up the Spout

Extends the BaseRichSpout abstract class

Set up the Bolt

Extends the BaseBasicBolt abstract class

Demo

Set up the Bolt

- Extend the BaseBasicBolt class
- Compute Gain/Loss signal
- Write data to file

```java
public void prepare(Map stormConf, TopologyContext context) {}

public void execute(Tuple input,BasicOutputCollector collector) {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}

public void cleanup() {}
```

Implementing the Bolt

```
public void prepare(Map stormConf, TopologyContext context) {}

public void execute(Tuple input,BasicOutputCollector collector) {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}

public void cleanup() {}
```

Initializing the Bolt

```
public void prepare(Map stormConf, TopologyContext context) {}

public void execute(Tuple input,BasicOutputCollector collector) {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}

public void cleanup() {}
```

Declare Fields

```java
public void prepare(Map stormConf, TopologyContext context) {}

public void execute(Tuple input,BasicOutputCollector collector) {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}

public void cleanup() {}
```

Process Input Tuple and Emit Values

```
public void prepare(Map stormConf, TopologyContext context) {}

public void execute(Tuple input,BasicOutputCollector collector) {}

public void declareOutputFields(OutputFieldsDeclarer declarer) {}

public void cleanup() {}
```

Cleanup on Bolt Shutdown

Stock Price Tracker

Set up a project

Use Maven to add Storm as a dependency

Set up the Spout

Extends the BaseRichSpout abstract class

Set up the Bolt

Extends the BaseBasicBolt abstract class

Run the topology

Build the topology

Submit to a local cluster

Running the Topology

Build the topology

Add the spout and bolt and connect them

Configure the topology

Specify the output file directory

Submit to a local cluster

Demo

Running the topology

Summary

Differentiate between real-time processing and batch processing

Understand the components of a Storm topology

Understand how data is represented

Implement a stock price tracker topology