

# Implementing Real-Time Analysis with Hadoop in Azure HDInsight

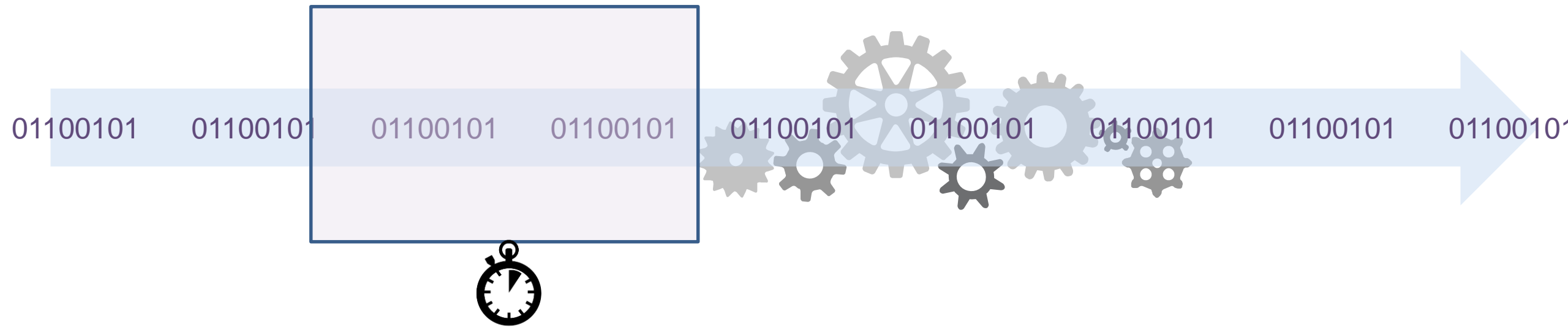
## 02 | Using Storm for Streaming Data



Graeme Malcolm | Snr Content Developer, Microsoft

- What is a Stream?
- What is Apache Storm?
- How is Storm Supported in Azure HDInsight?
- What is a Storm Topology?
- How is Event Data Defined?
- How Does Storm Distribute Stream Processing?
- How Does Storm Guarantee Message Processing?
- How Do I Aggregate Data in a Stream?

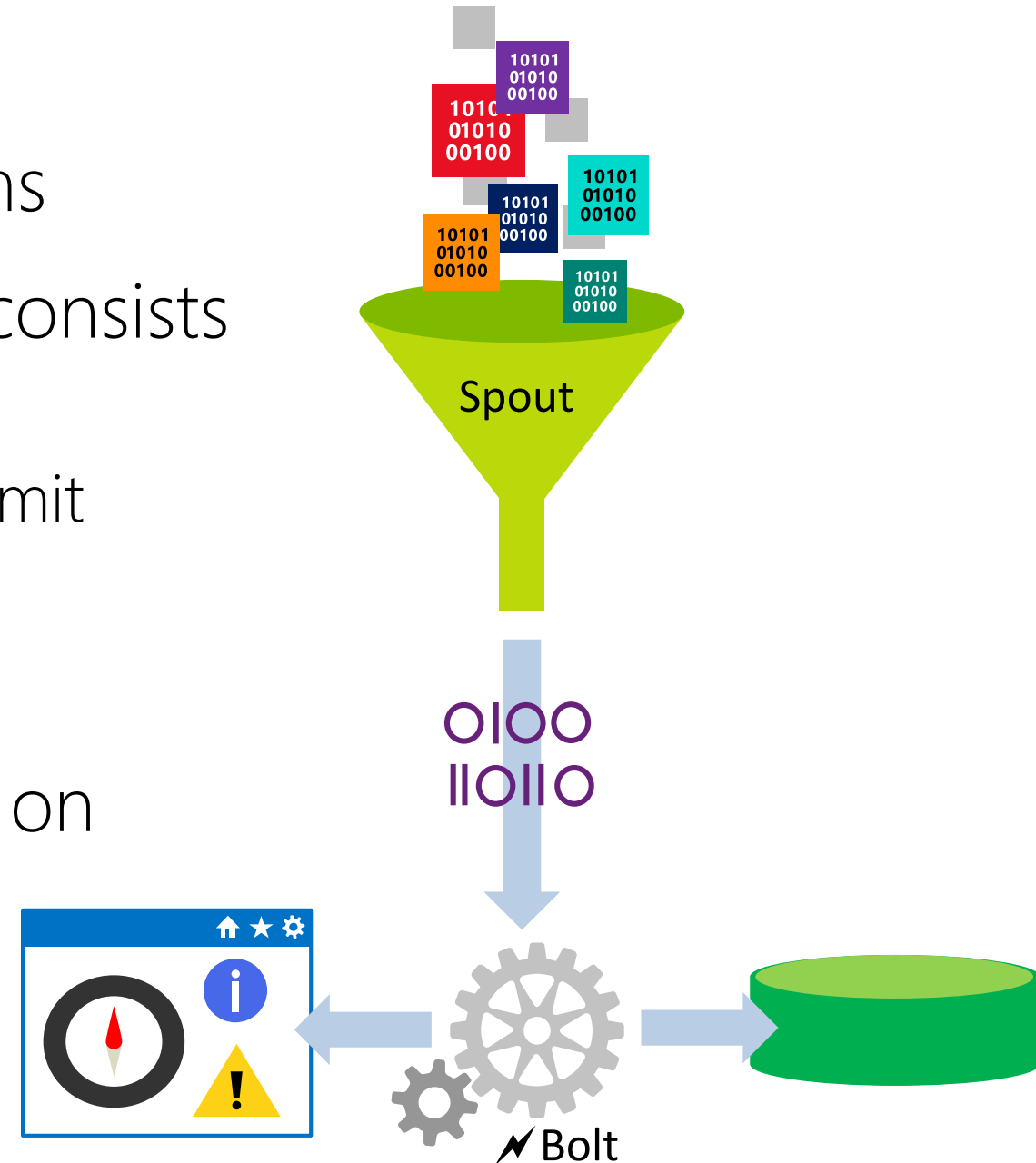
What is a Stream?



- A *unbounded* sequence of event data
- Stream processing is *continuous*
- Aggregation is based on temporal *windows*

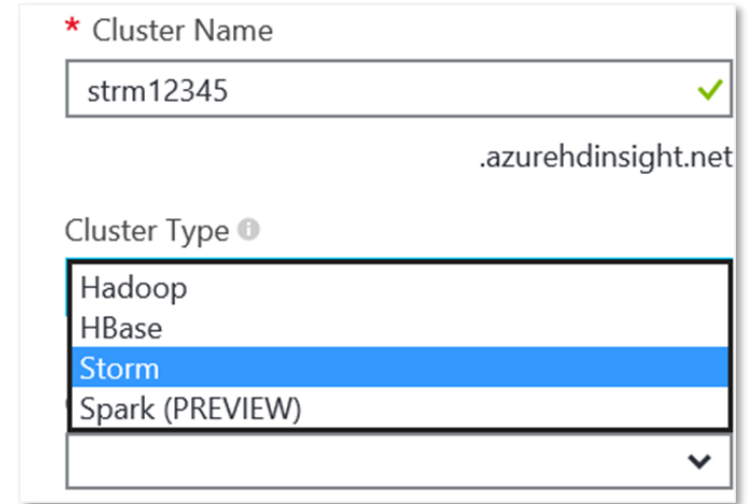
What is Apache Storm?

- An event processor for data streams
- Defines a streaming *topology* that consists of:
  - *Spouts*: Consume data sources and emit streams that contain *tuples*
  - *Bolts*: Operate on tuples in streams
- Storm topologies run continuously on streams of data
  - Real-time monitoring
  - Event aggregation and logging



How is Storm Supported in Azure HDInsight?

- HDInsight supports an **Storm** cluster type
  - Choose Cluster Type in the Azure Portal
- Can be provisioned in a virtual network



The screenshot shows a portion of the Azure Portal's cluster creation interface. At the top, there is a field for 'Cluster Name' with a red asterisk indicating it is required. The text 'strm12345' is entered in the field, and a green checkmark is visible to the right. Below this, the text '.azurehdinsight.net' is partially visible. Underneath, there is a section for 'Cluster Type' with an information icon. A dropdown menu is open, showing four options: 'Hadoop', 'HBase', 'Storm' (which is highlighted with a blue background), and 'Spark (PREVIEW)'. A downward arrow is visible at the bottom right of the dropdown menu.

\* Cluster Name

strm12345 ✓

.azurehdinsight.net

Cluster Type ⓘ

- Hadoop
- HBase
- Storm
- Spark (PREVIEW)



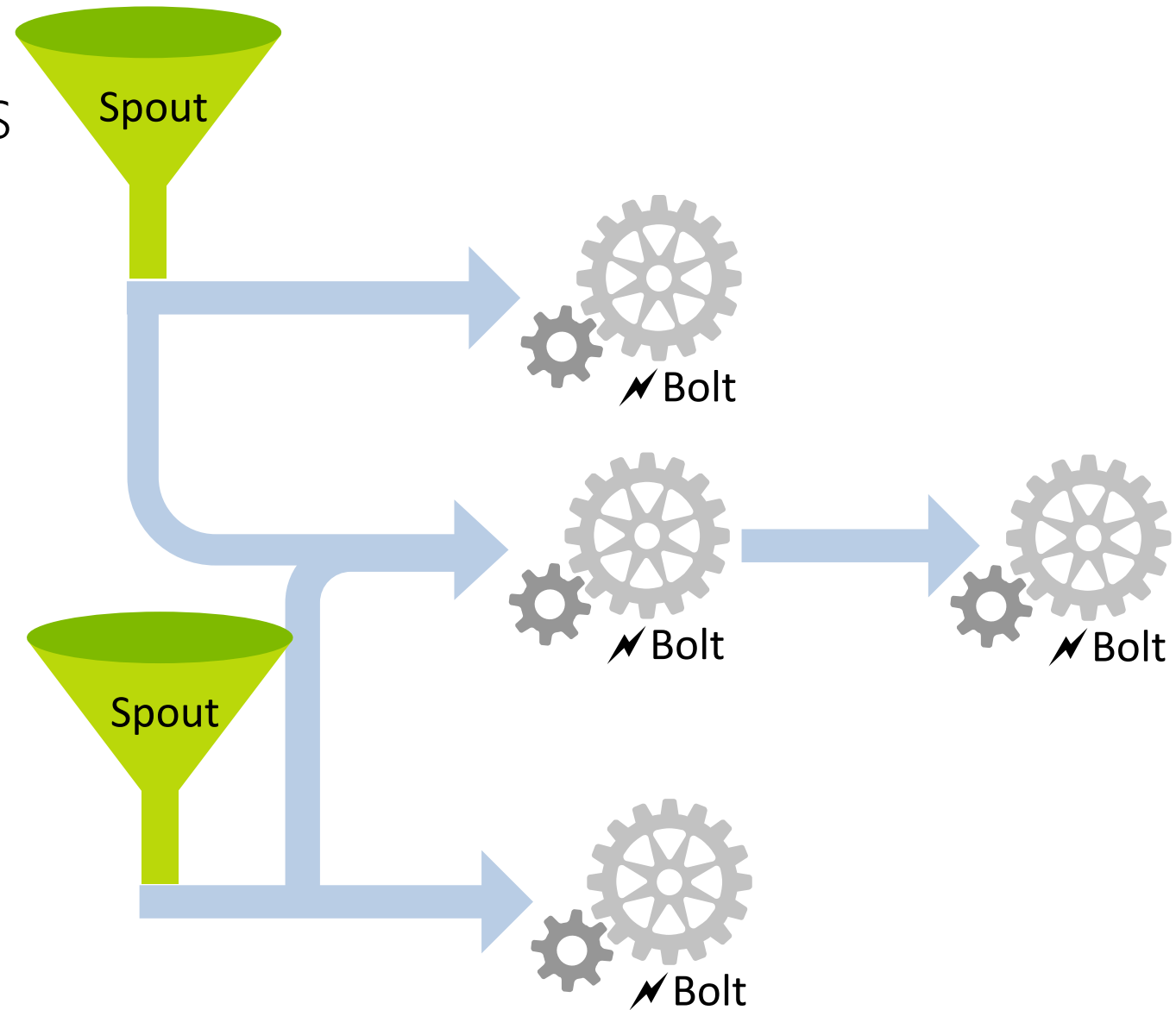
# DEMO

---

Provisioning a Storm Cluster

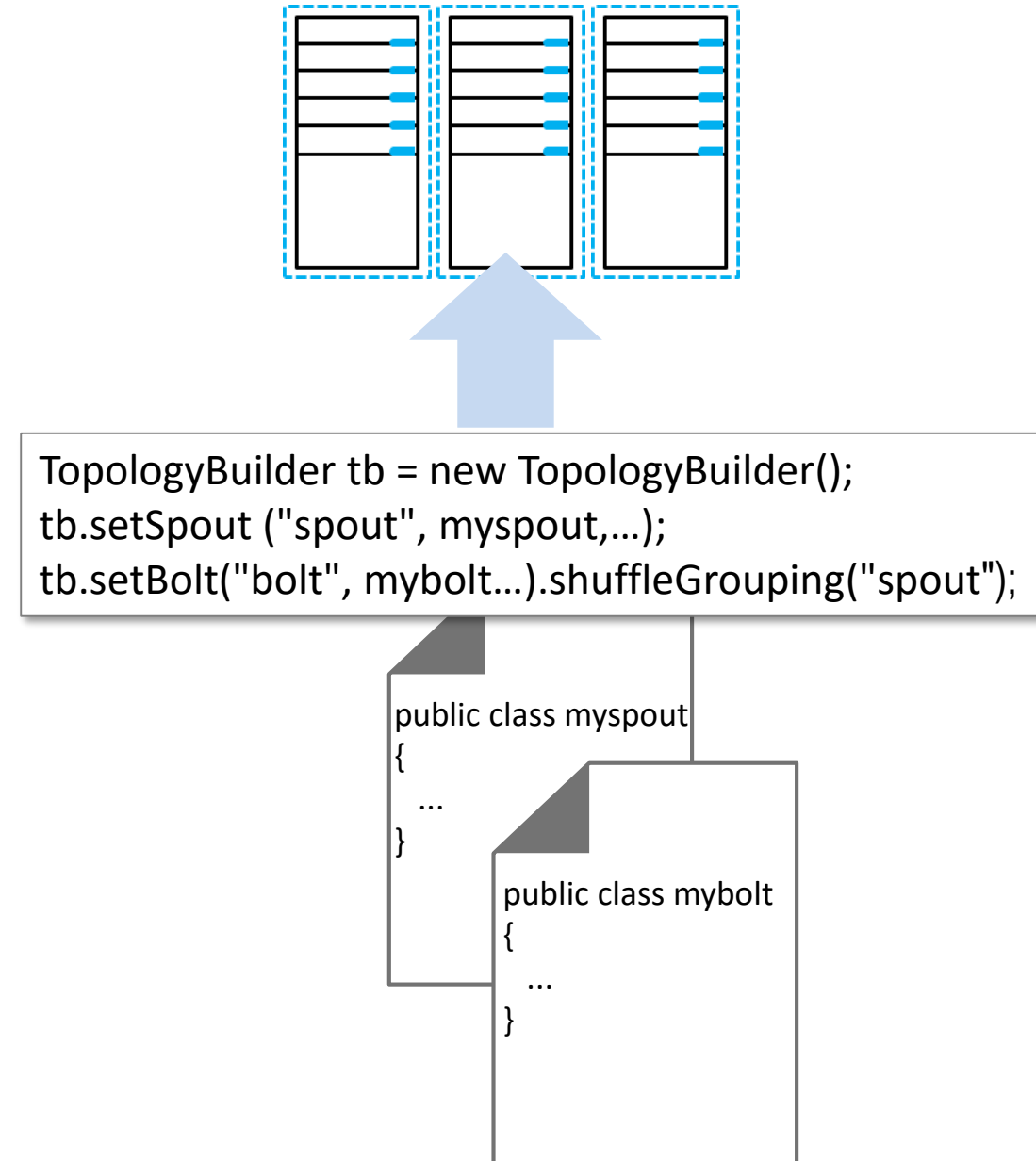
What is a Storm Topology?

- Spouts emit tuples in streams
- Spouts can emit multiple streams
- Bolts process tuples
- Bolts can also emit tuples
- There can be multiple spouts and bolts in a topology
- Bolts can process multiple streams



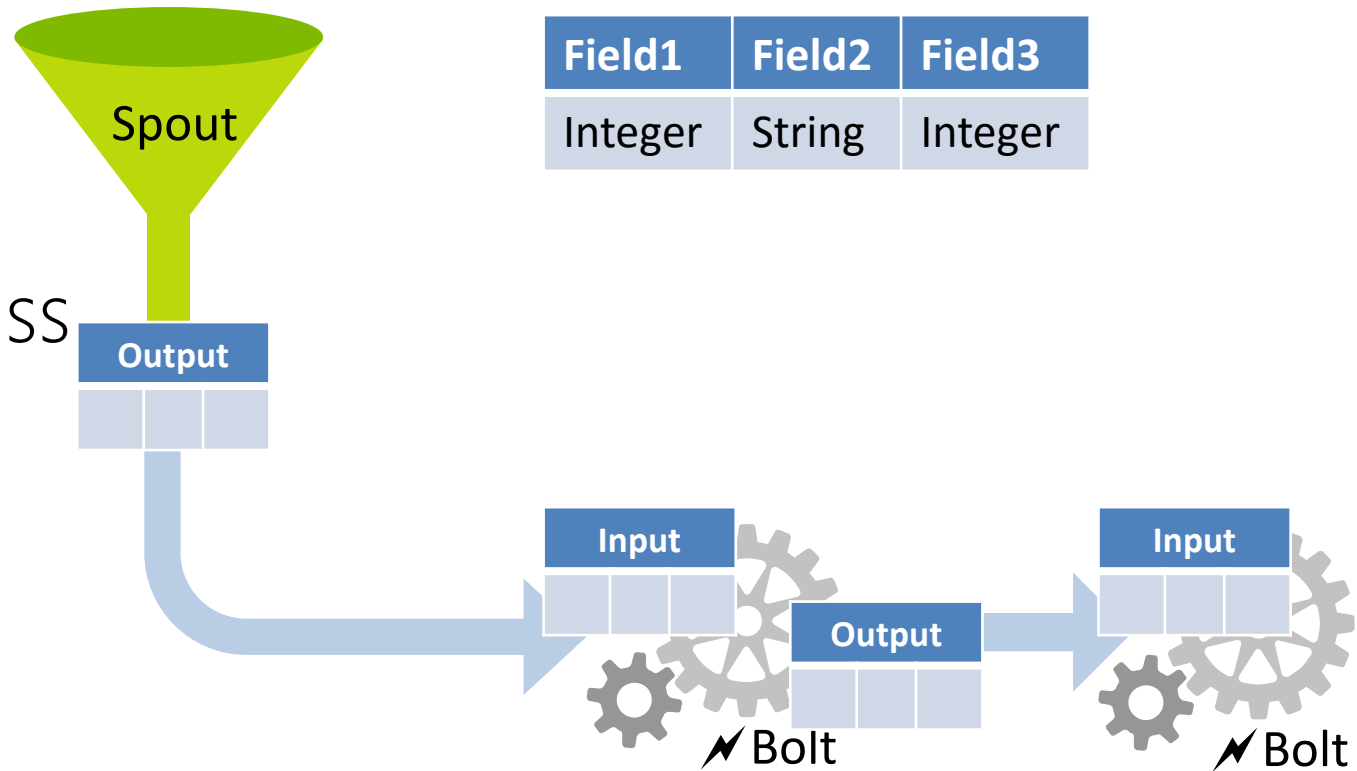
How do I Create a Topology?

- Implement **Spout** and **Bolt** classes
  - Native language of Storm is Java
  - Microsoft SCP.NET package enables development in C#
- Use a **TopologyBuilder** class to connect the components
- Build and package the code, and submit the topology to a Storm cluster



How is Event Data Defined?

- Declare schema for each stream in each component
- Java **OutputFieldsDeclarer** class defines output schema for a stream
- Microsoft SCP.NET class templates include input and output schema declarations for spouts and bolts



# DEMO

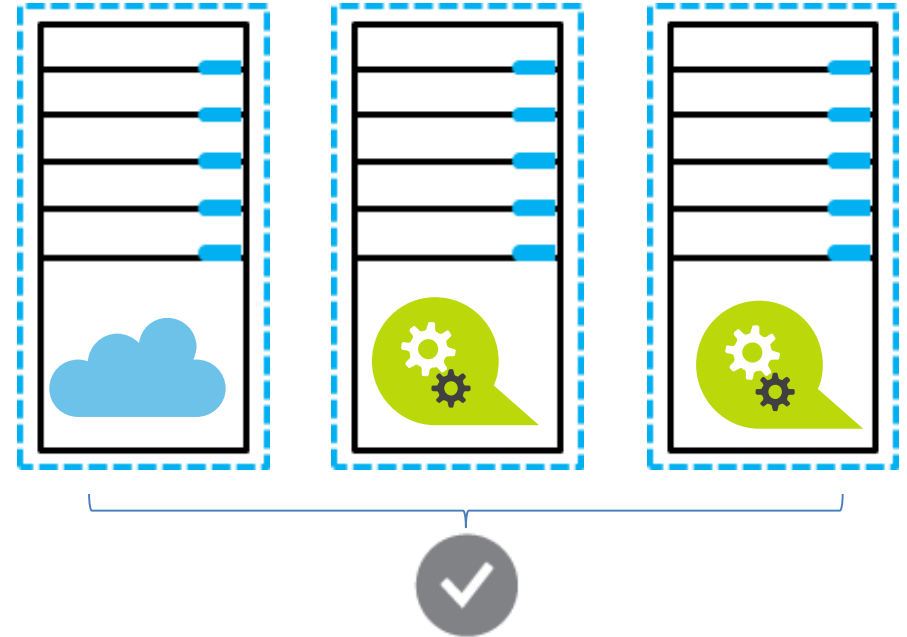
---

Creating a Storm Topology with C#



How Does Storm Distribute Stream Processing?

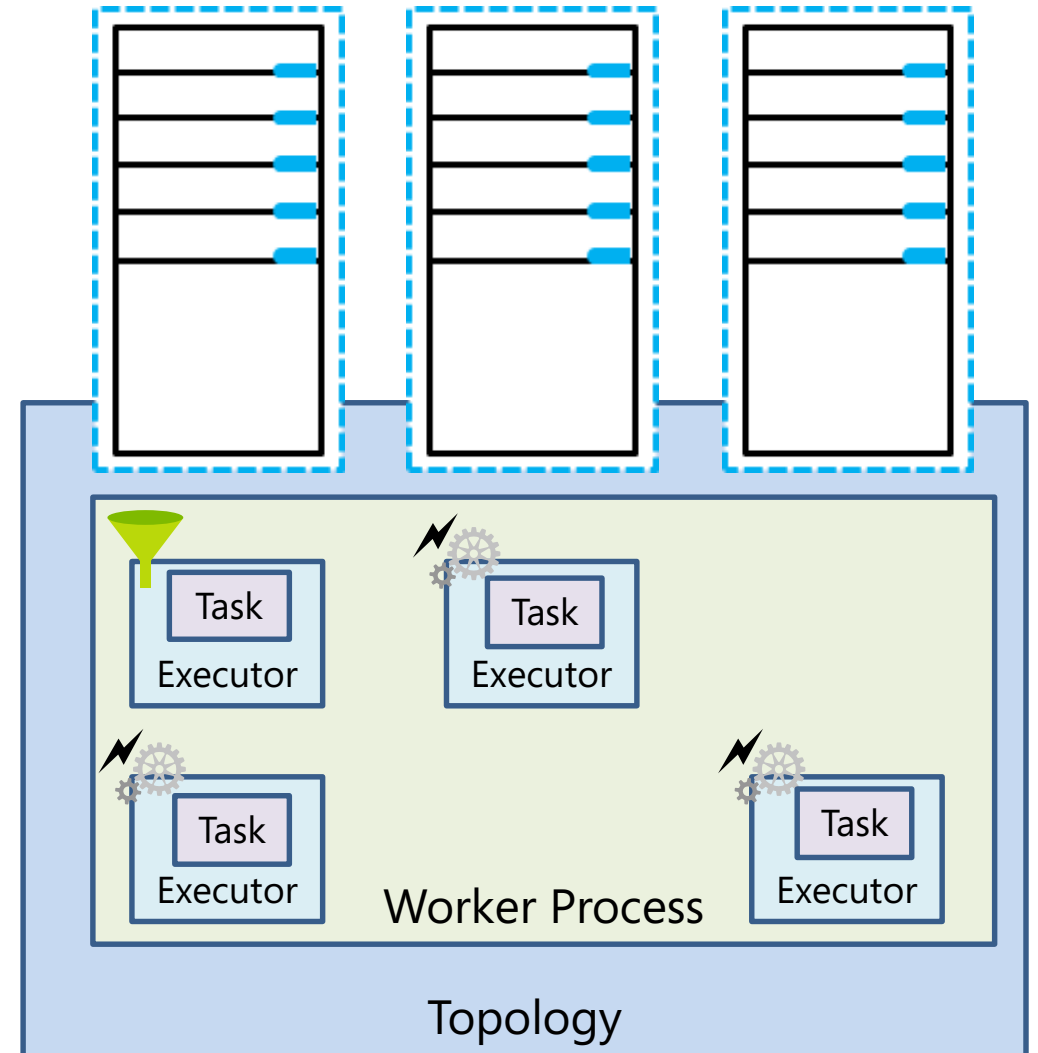
- Master node runs *Nimbus*
  - Assigns processing across the cluster
- Worker nodes run *Supervisor*
  - Manages processing on the node
- Cluster coordination is managed using *Zookeeper*
  - Apache project for distributed processing



- A topology has one or more *worker processes*
- A worker process spawns one or more *executors* (threads) per component
  - Set using parallelism hint

```
TopologyBuilder tb = new TopologyBuilder();  
tb.setSpout("spout", myspout, 1, ...);  
tb.setBolt("bolt", mybolt, 3, ...).shuffleGrouping("spout");
```

- Each executor runs one or more *task*



- Use stream groupings to determine affinity between tasks
  - *Shuffle* grouping

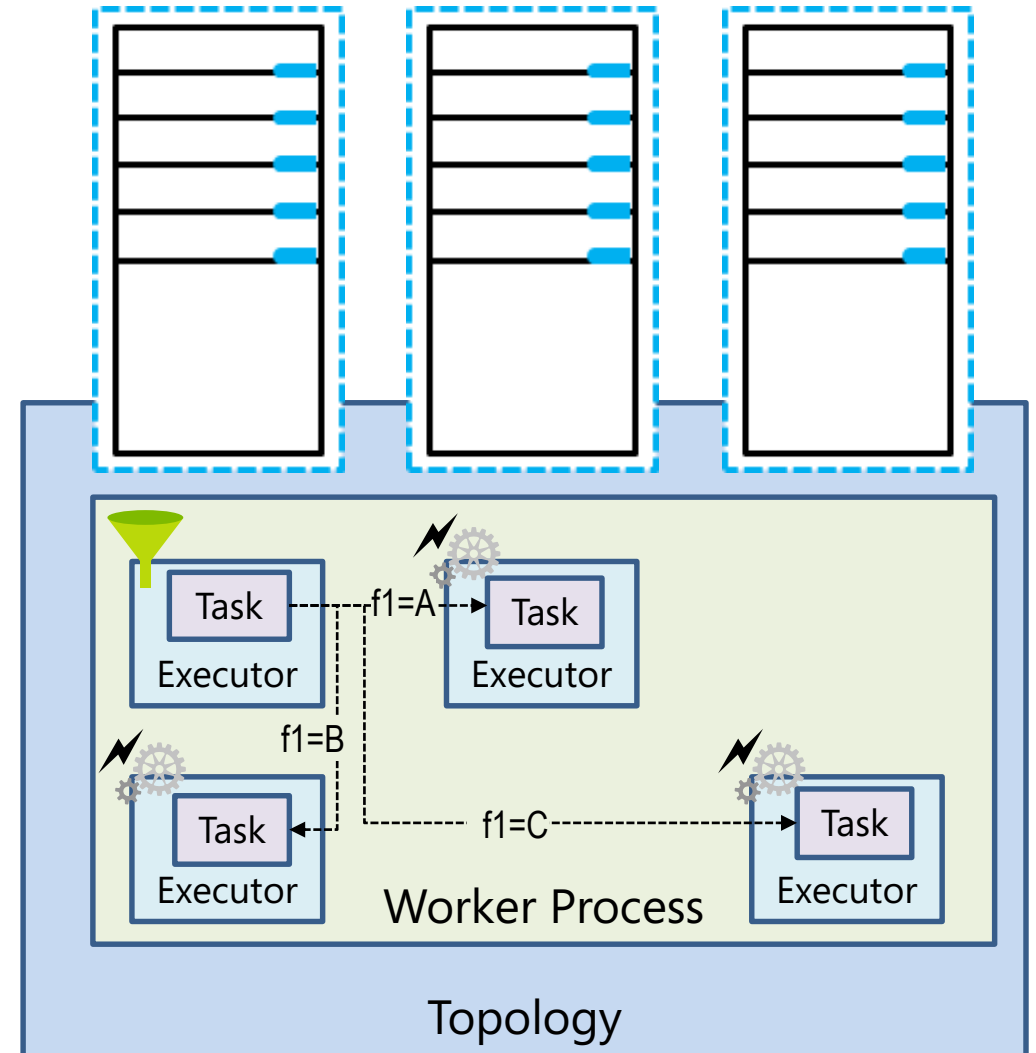
```
TopologyBuilder tb = new TopologyBuilder();  
tb.setSpout("spout", myspout, 1, ...);  
tb.setBolt("bolt", mybolt, 3, ...).shuffleGrouping("spout");
```

- *Fields* grouping

```
TopologyBuilder tb = new TopologyBuilder();  
tb.setSpout("spout", myspout, 1, ...);  
tb.setBolt("bolt", mybolt, 3, ...).fieldsGrouping("spout", "f1");
```

- Others

- *All, Global, ...*



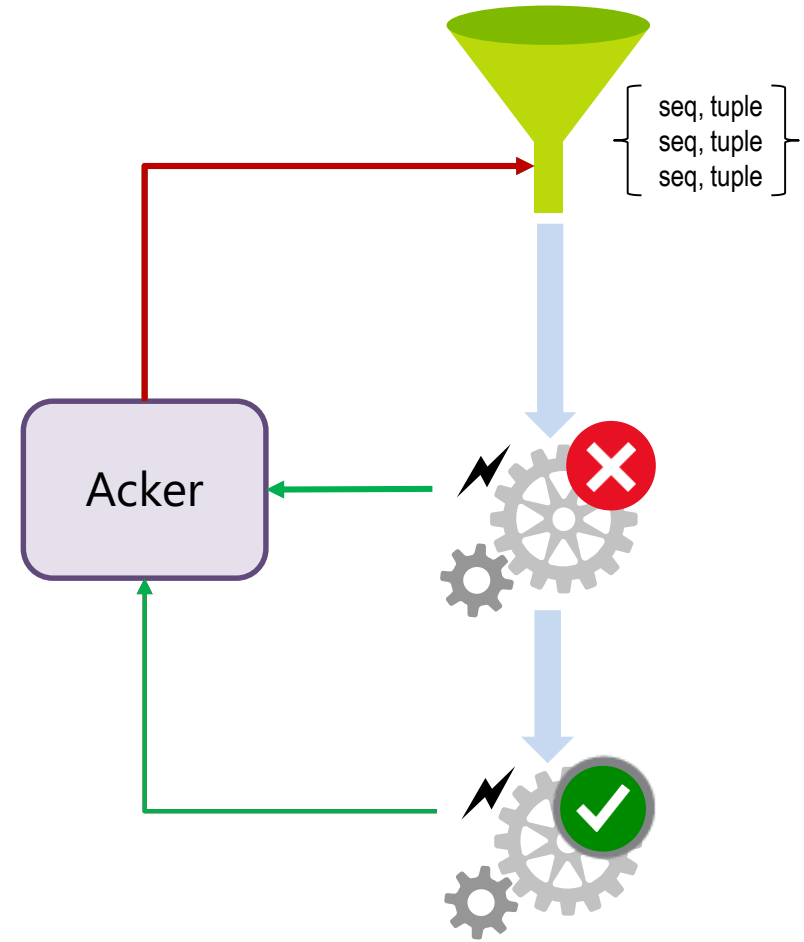
# DEMO

---

Using the Parallelism Hint

How Does Storm Guarantee Message Processing?

- Non-Transactional (no Ack)
  - Enforces *at most once* semantics
  - Simplest programming model
  - Possible data loss
- Non-Transactional (with Ack)
  - Enforces *at least once* semantics
  - Requires explicit retry logic
- Transactional
  - Enforces *exactly once* semantics
  - Works well for batches
  - Use **TransactionalTopologyBuilder**
  - Implement a *committer* bolt



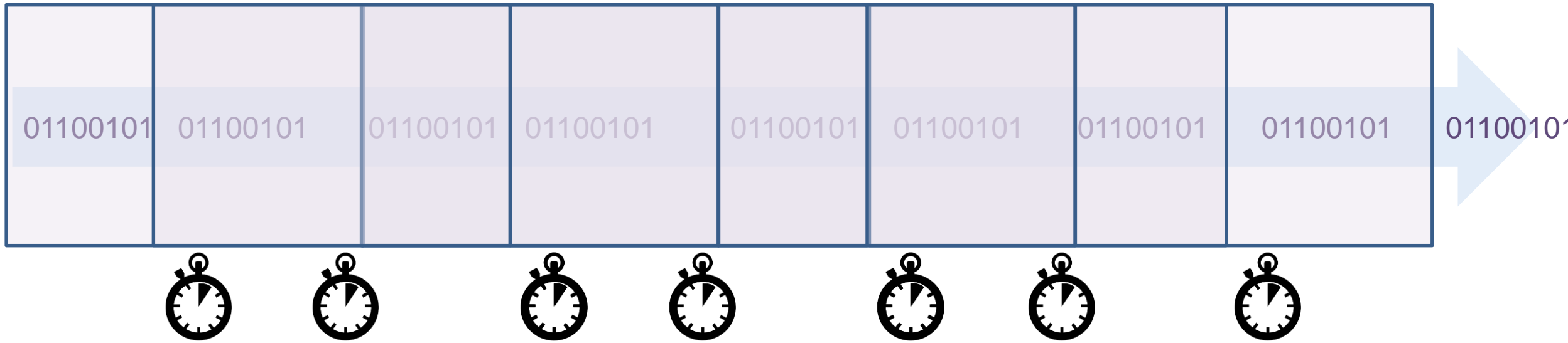
# DEMO

---

Implementing Guaranteed Message Processing

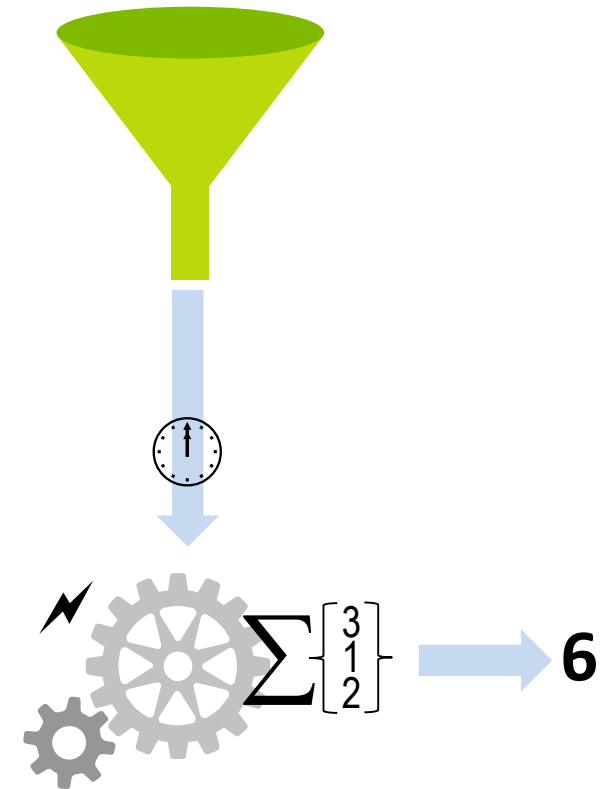


How Do I Aggregate Data in a Stream?



- Events are aggregate within temporal *windows*
- Use a *tumbling window* to aggregate events in a fixed timespan
  - For example: every hour, count the events in the preceding hour
- Use a *sliding window* to aggregate events in overlapping timespans
  - For example: every 10 minutes, count the events in the preceding hour

- Cache field values from each tuple
- Configure a *Tick Tuple* for the window duration
- On each tick, start a new window:
  - For a tumbling window:
    - Aggregate cached fields
    - Delete all cached fields
  - For a sliding window
    - Delete stale fields
    - Aggregate remaining fields



# DEMO

---

Implementing a Sliding Window

- What is a Stream?
- What is Apache Storm?
- How is Storm Supported in Azure HDInsight?
- What is a Storm Topology?
- How is Event Data Defined?
- How Does Storm Distribute Stream Processing?
- How Does Storm Guarantee Message Processing?
- How Do I Aggregate Data in a Stream?



# Microsoft

©2014 Microsoft Corporation. All rights reserved. Microsoft, Windows, Office, Azure, System Center, Dynamics and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.