APACHE STORM

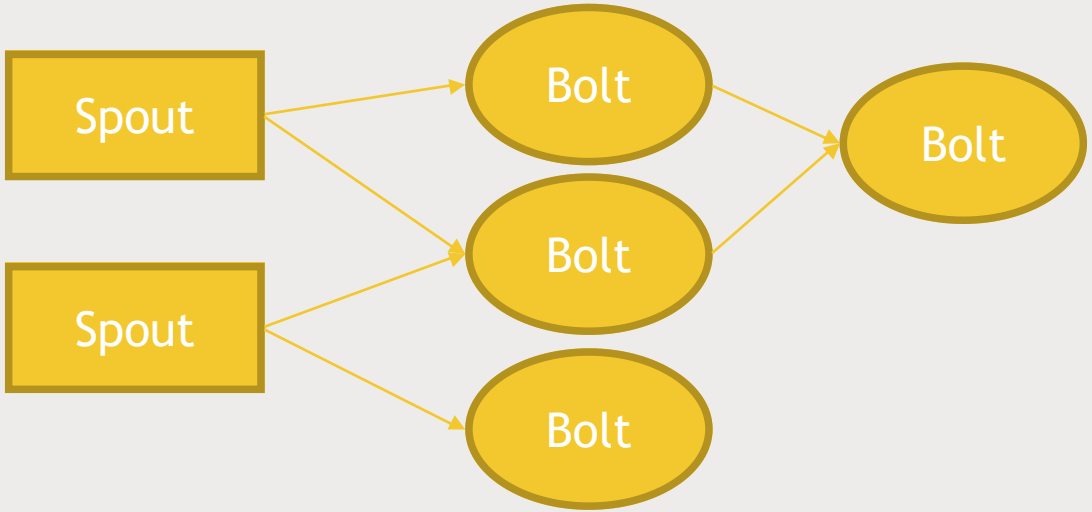
Real-time stream processing

What is Storm?

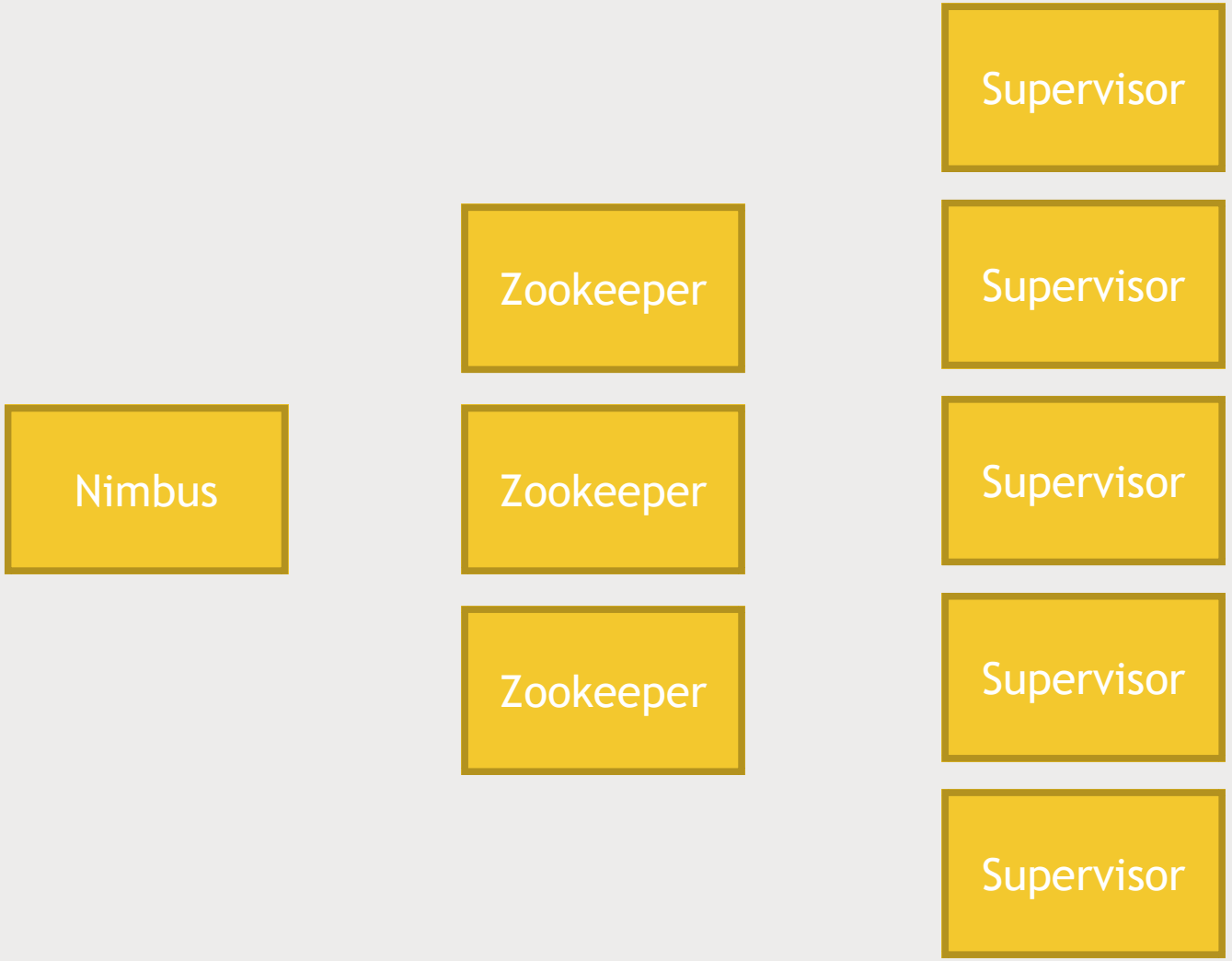
- Another framework for processing continuous streams of data on a cluster
  - *Can run on top of YARN (like Spark)*
- Works on individual events, not micro-batches (like Spark Streaming does)
  - *If you need sub-second latency, Storm is for you*

Storm terminology

- A *stream* consists of *tuples* that flow through…
- *Spouts* that are sources of stream data (Kafka, Twitter, etc.)
- *Bolts* that process stream data as it's received
  - *Transform, aggregate, write to databases / HDFS*
- A *topology* is a graph of spouts and bolts that process your stream



Storm architecture



Developing Storm applications

- Usually done with Java
  - *Although bolts may be directed through scripts in other languages*
- Storm Core
  - *The lower-level API for Storm*
  - *“At-least-once” semantics*
- Trident
  - *Higher-level API for Storm*
  - *“Exactly once” semantics*
- Storm runs your applications “forever” once submitted – until you explicitly stop them

Storm vs. Spark Streaming

- There’s something to be said for having the rest of Spark at your disposal
- But if you need truly real-time processing (sub-second) of events as they come in, Storm’s your choice
- Core Storm offers “tumbling windows” in addition to “sliding windows”
- Kafka + Storm seems to be a pretty popular combination

Let’s Play

- We’ll run the WordCount topology example and examine it.