Introduction to Storm

LI Tao

Hong Kong University of Science and Technology

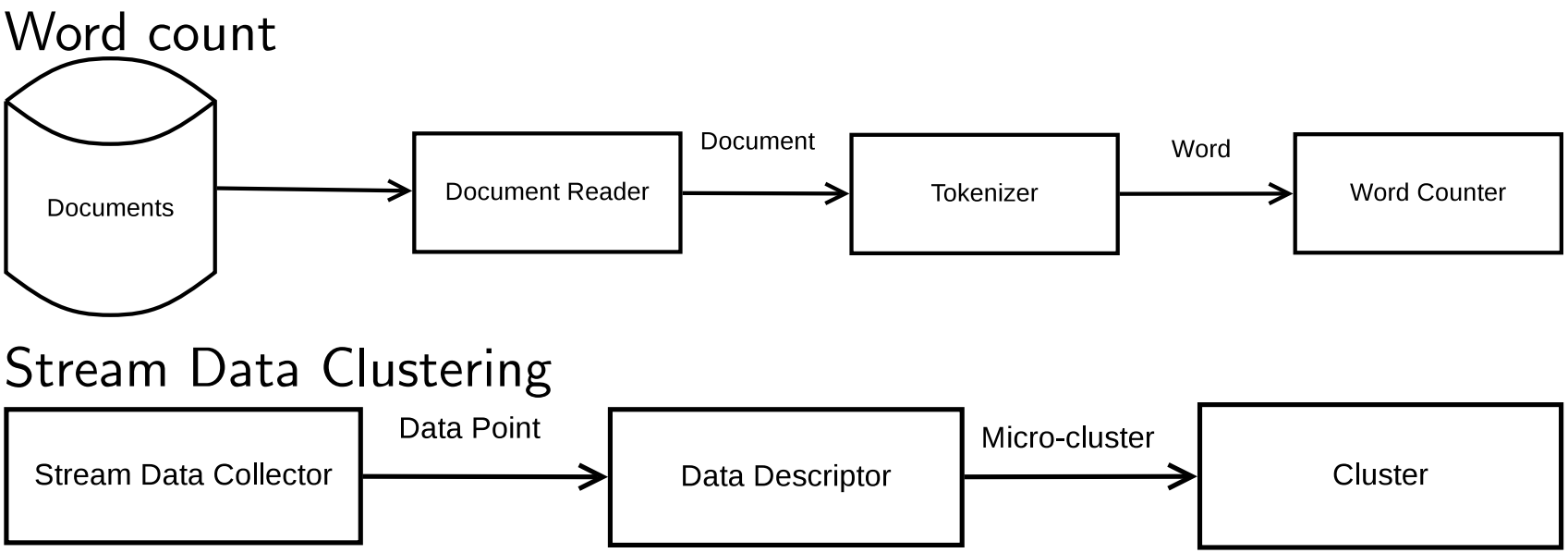
tliab@ust.hk

November 15, 2013

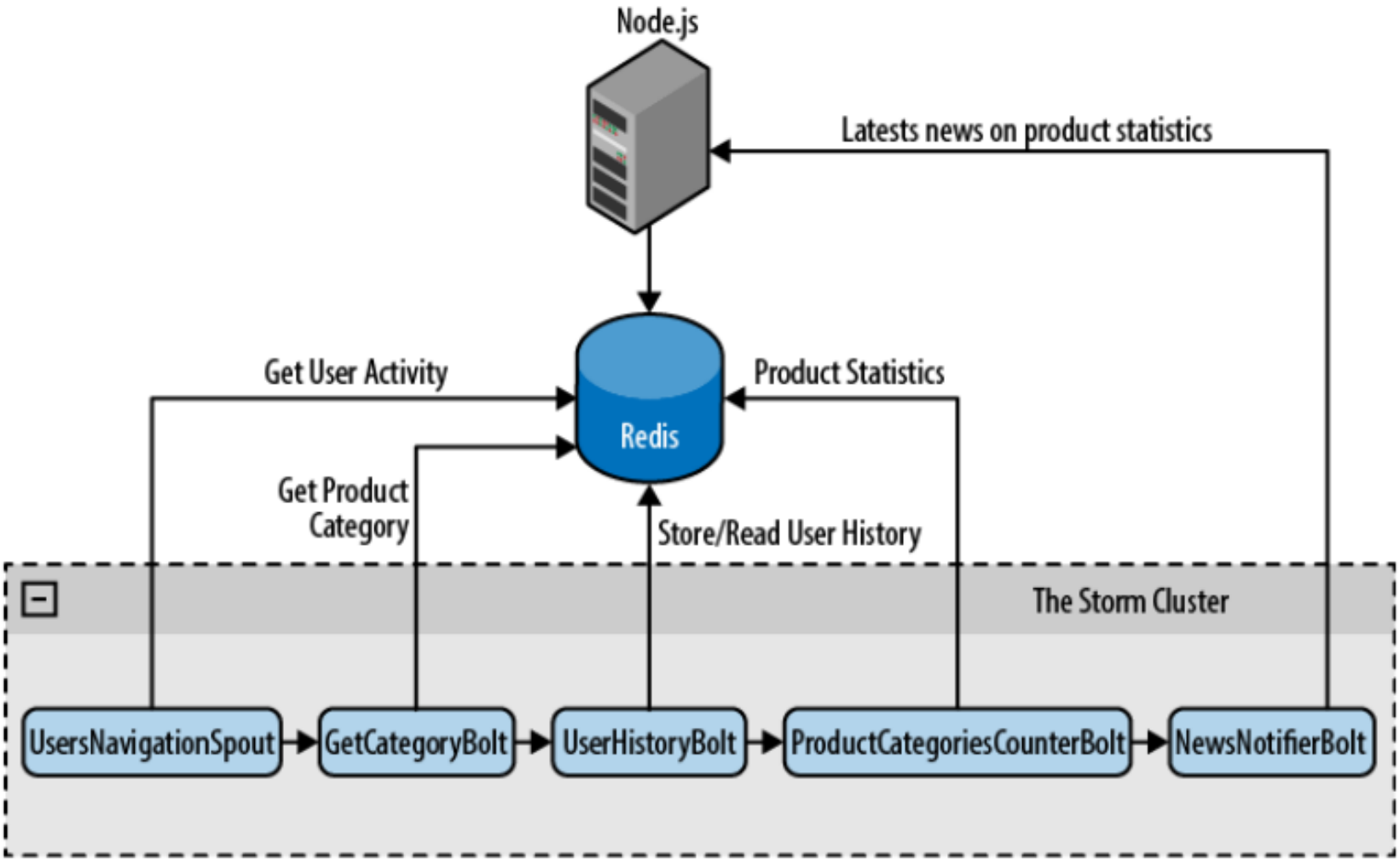
Why Storm?

- **Simple and Beautiful**: simple topology, easy to convert from existing single thread application.
- **Reliable**: all messages are guaranteed to be processed at least once.
- **Scalable**: all you need to do in order to scale is add more machines to the cluster. Storm will automatically reassign tasks to new machines as they become available.

Examples


Word count
Documents
Document Reader
Document
Tokenizer
Word
Word Counter
Stream Data Clustering
Stream Data Collector
Data Point
Data Descriptor
Micro-cluster
Cluster

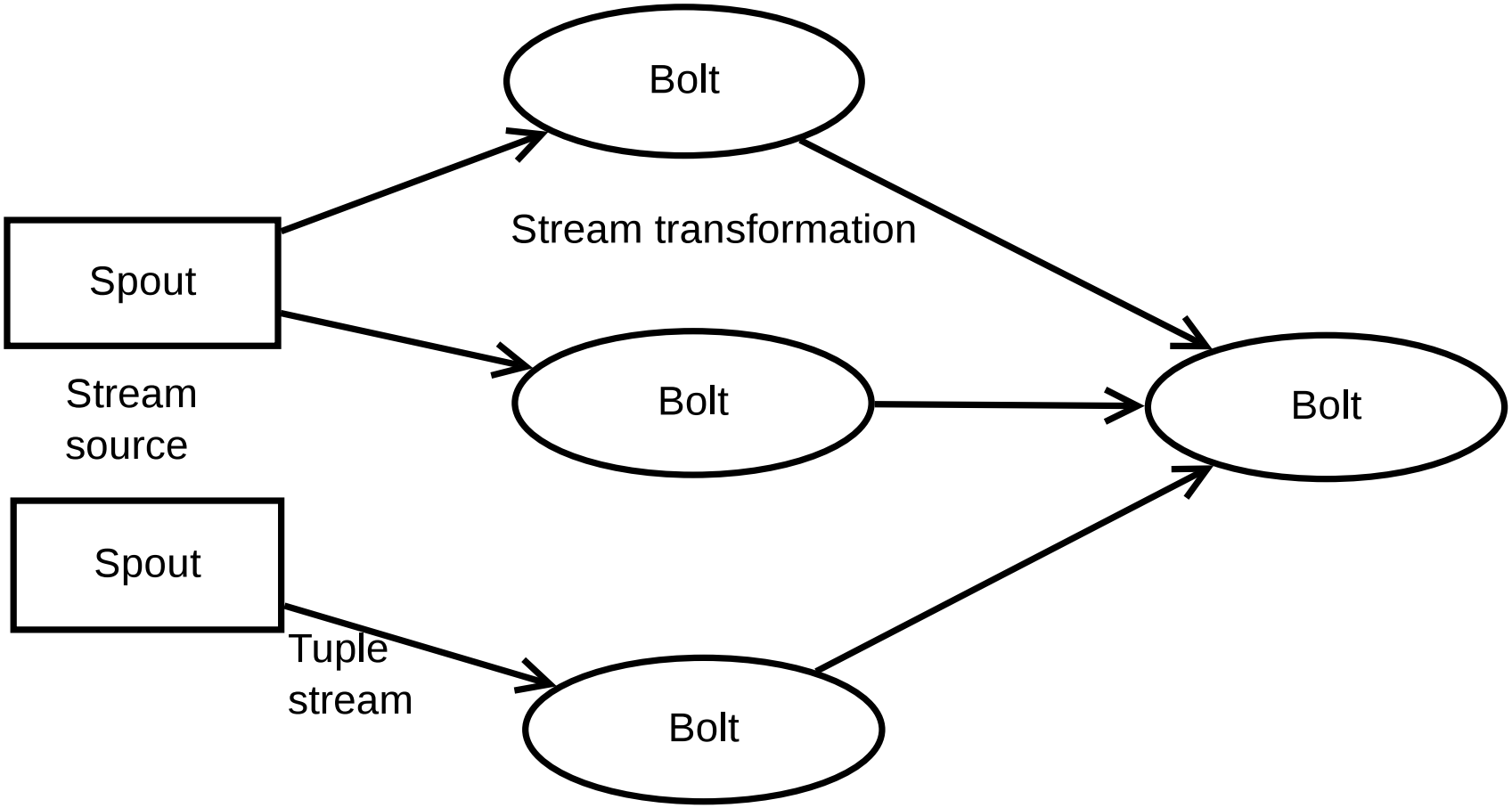
Real World Example



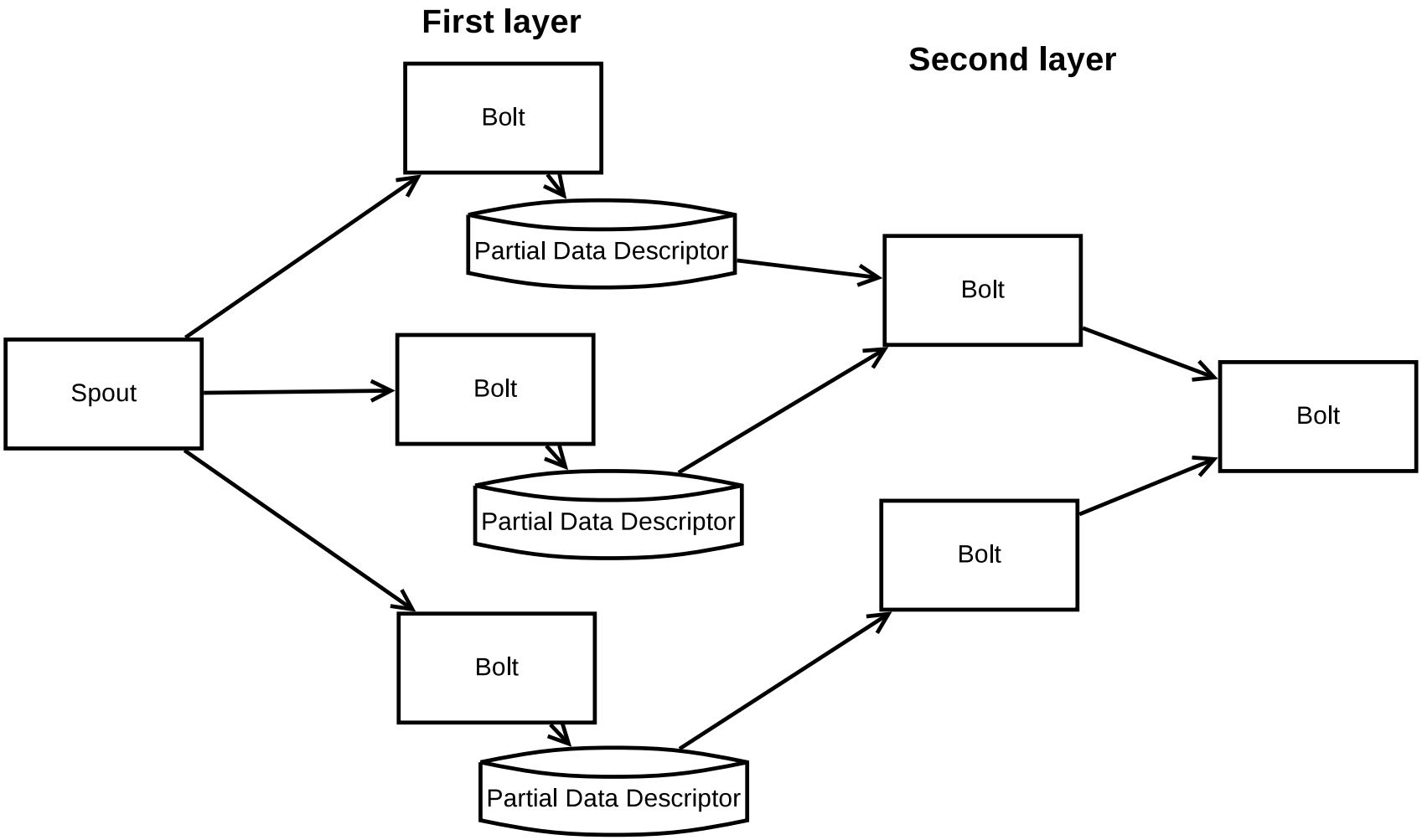
The common things on stream data

- A data generator, emit data point one by one.
- Several layers of logic to process/transform data.
- Each layer emit processed data point to the next layer.

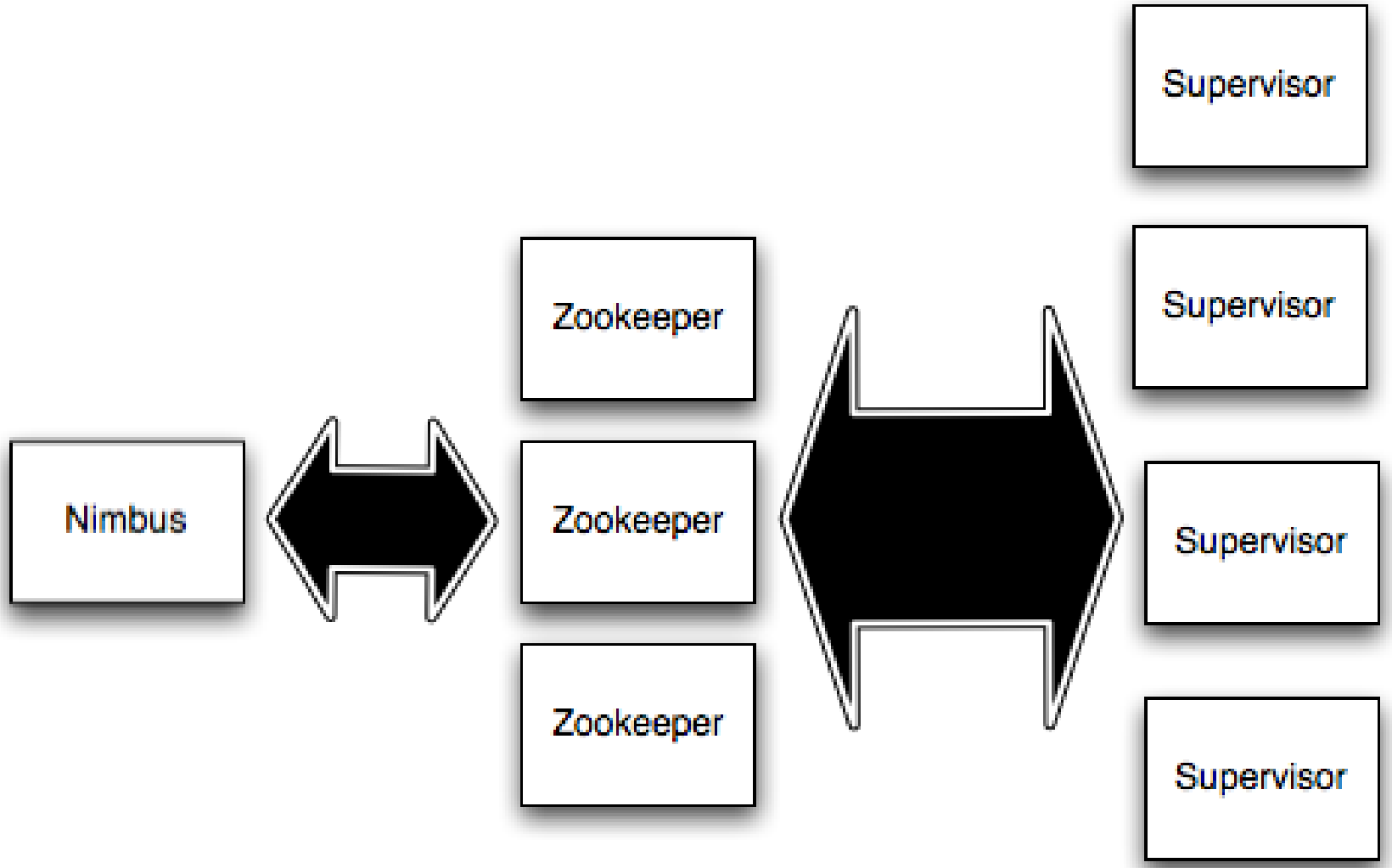
The Structure of Storm



- Spout: Information source, emitting stream of tuples.
- Bolt: Logic to process or transform tuples.


First layer
Second layer
Bolt
Partial Data Descriptor
Bolt
Spout
Bolt
Bolt
Partial Data Descriptor
Bolt
Bolt
Partial Data Descriptor

physical structure



Lifecycle of Spout

- `open()` the first method called in any spout
- `nextTuple()`: emit values to be processed by the bolts.
- `ack(msgId)`: called after a tuple is successfully processed
- `fail(msgId)`: called when a bolt fail to process a tuple.

- Bolts are created on the client machine, serialized into the topology, and submitted to the master machine of cluster
- The cluster launches workers that deserialized the bolt, call prepare on it, and then start processing tuples
- The most important method in bolt is `execute()`

Thank you