Storm

The Real-Time Layer Your Big Data's Been Missing

Dan Lynn
dan@fullcontact.com
@danklynn

FullContact

Keeps Contact Information Current and Complete

Based in **Denver, Colorado**

Advisors to FullContact
include some of the leading pioneers in internet startups.

techstars

FullContact is
a TechStars 2011 Boulder Graduate.

CTO & Co-Founder
dan@fullcontact.com
@danklynn

FullContact

Turn Partial Contacts Into **Full** Contacts

FullContact

Your data is **old**

FullContact

Old data is **confusing**

FullContact

First, there was the spreadsheet

Next, there was SQL

FullContact

Then, there wasn't SQL

FullContact

Batch Processing

Stale Data

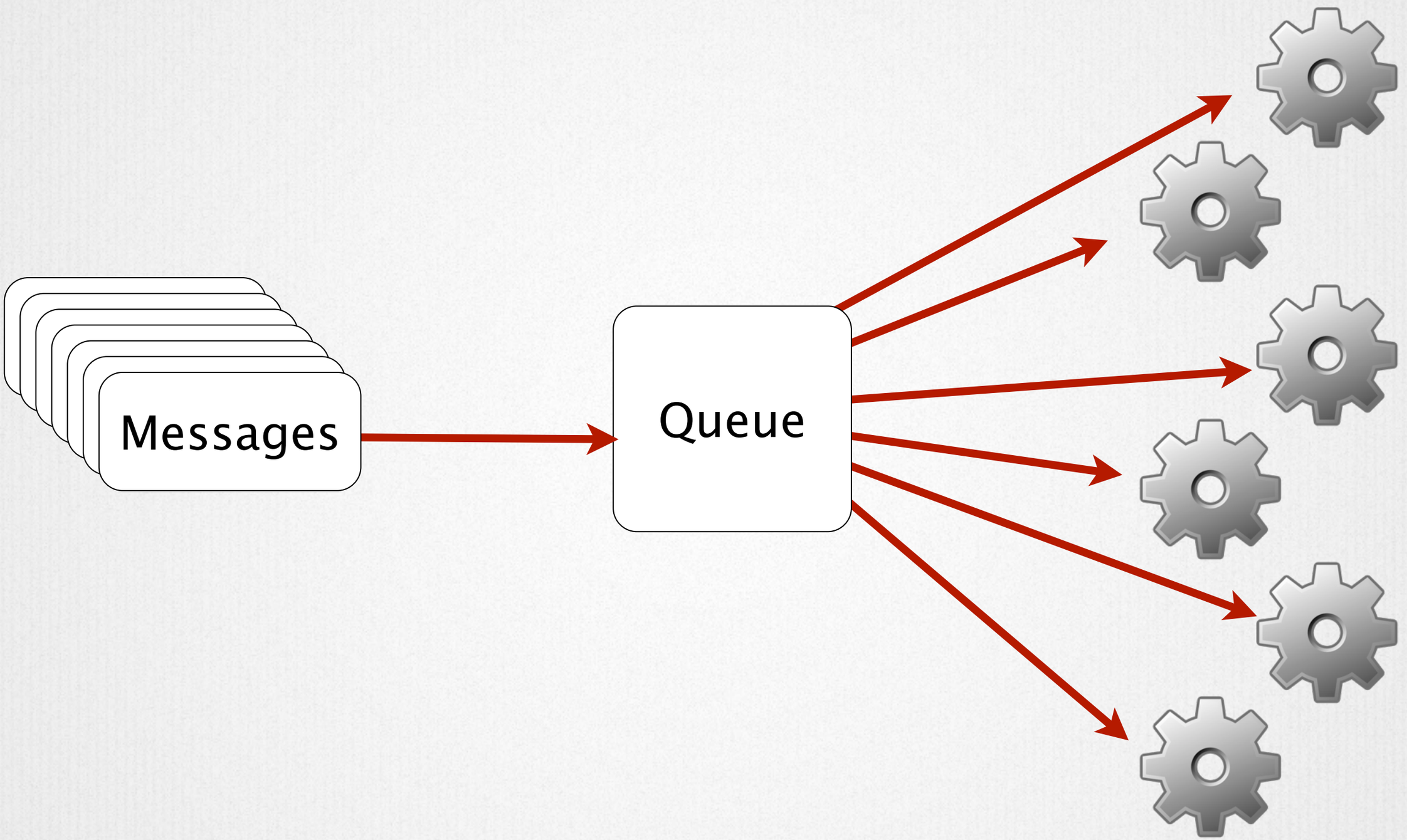
FullContact

Streaming Computation

FullContact

Queues / Workers

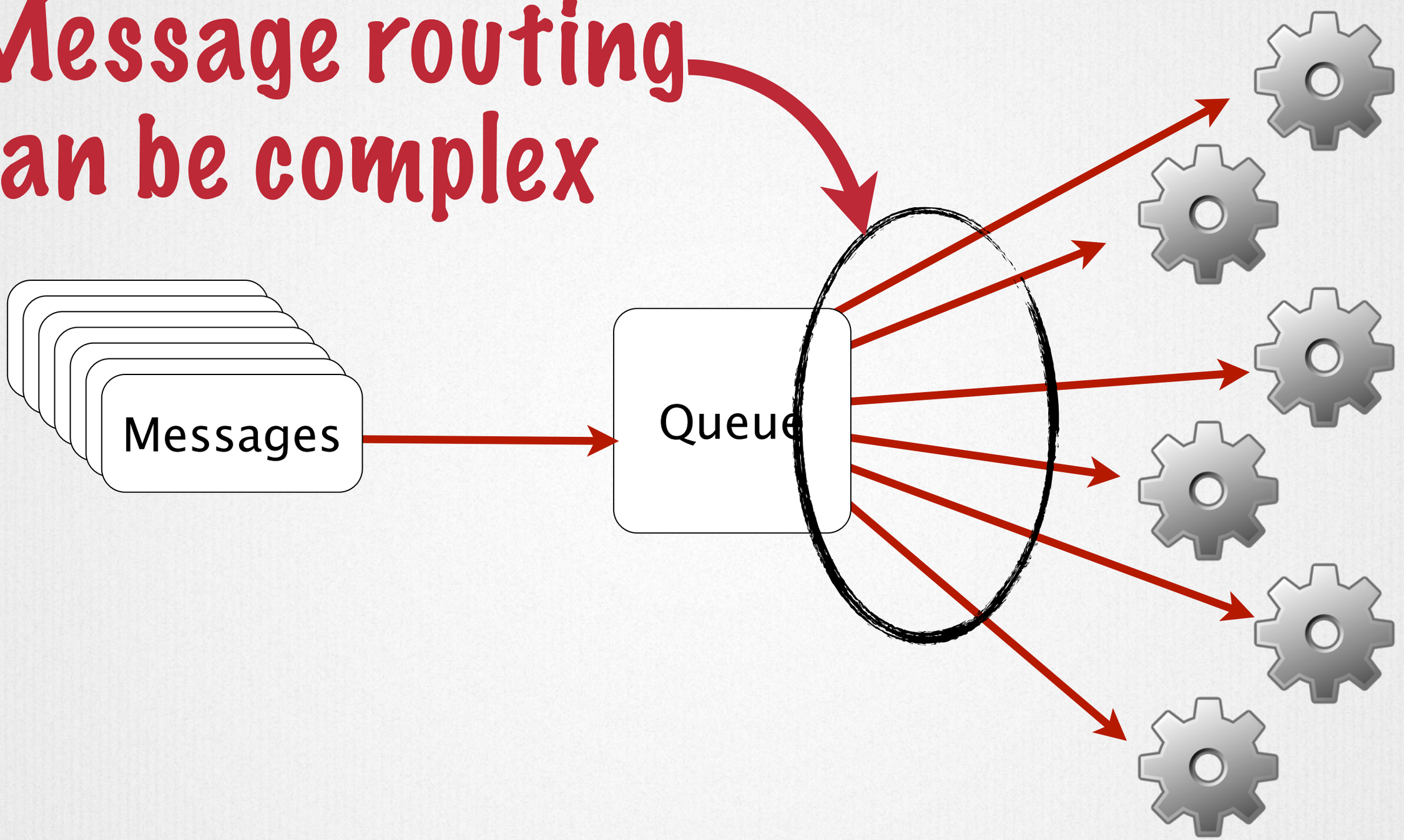
Queues / Workers



FullContact

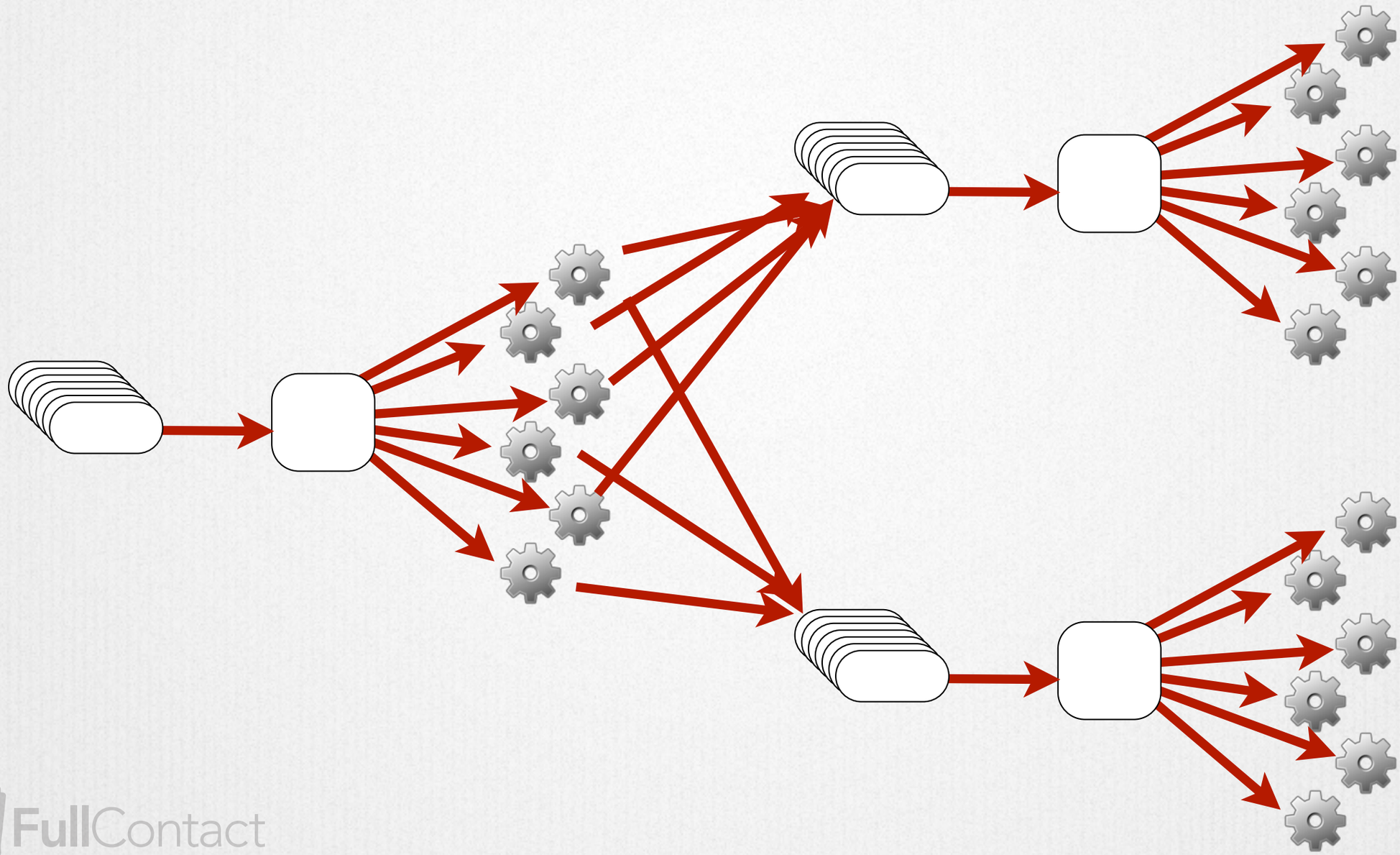
Queues / Workers

Message routing can be complex



FullContact

Queues / Workers

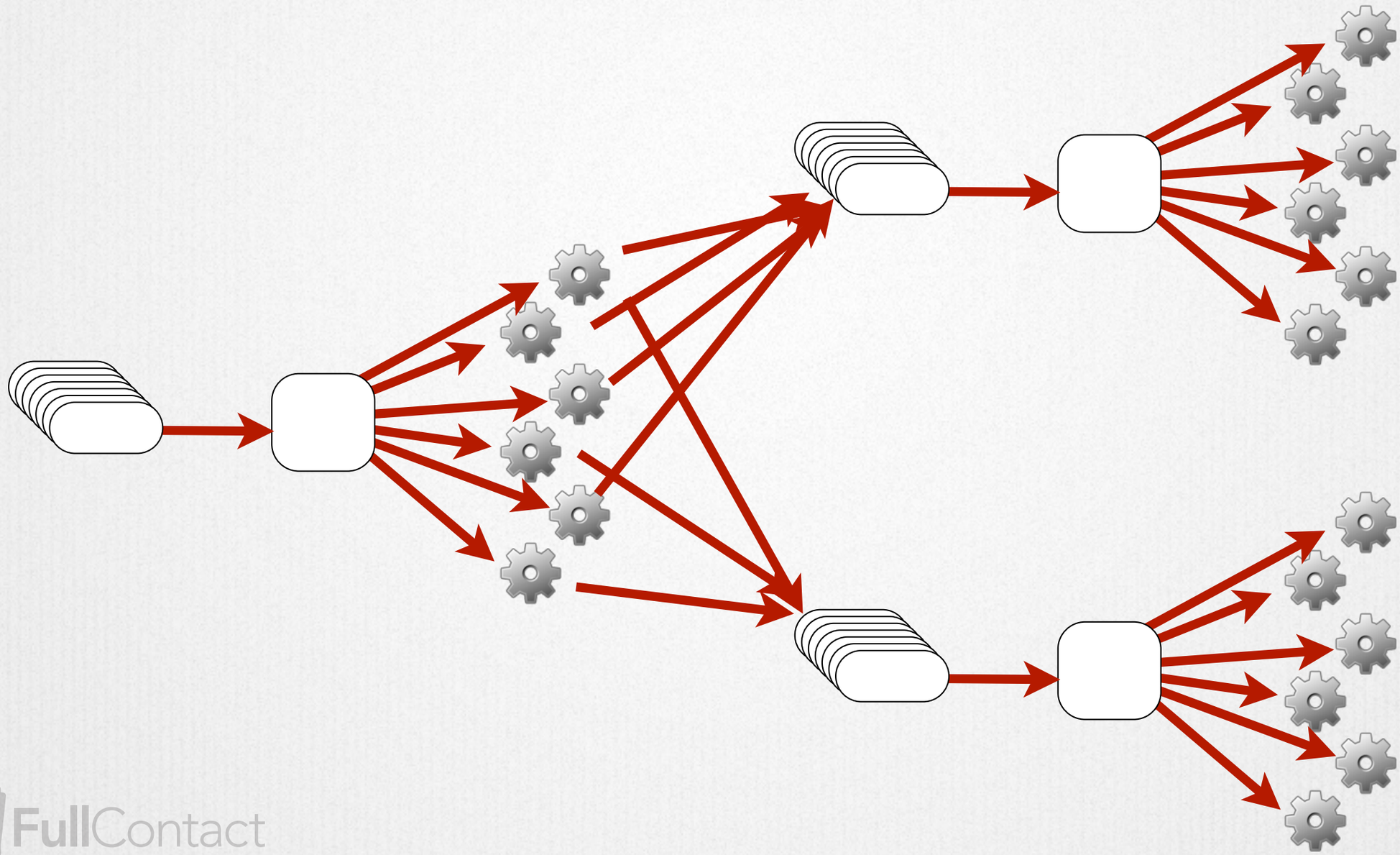


Brittle

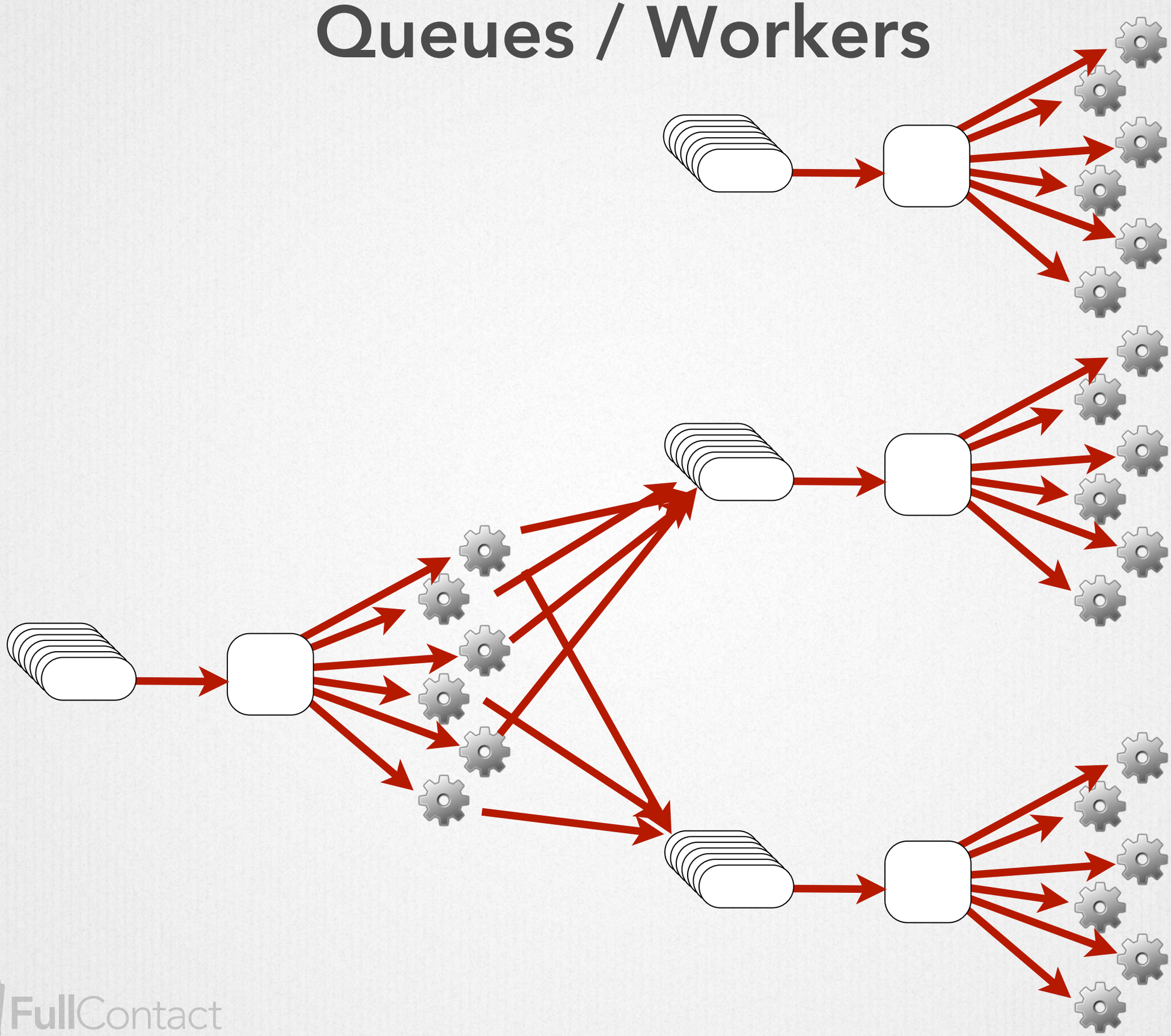
FullContact

Hard to scale

Queues / Workers

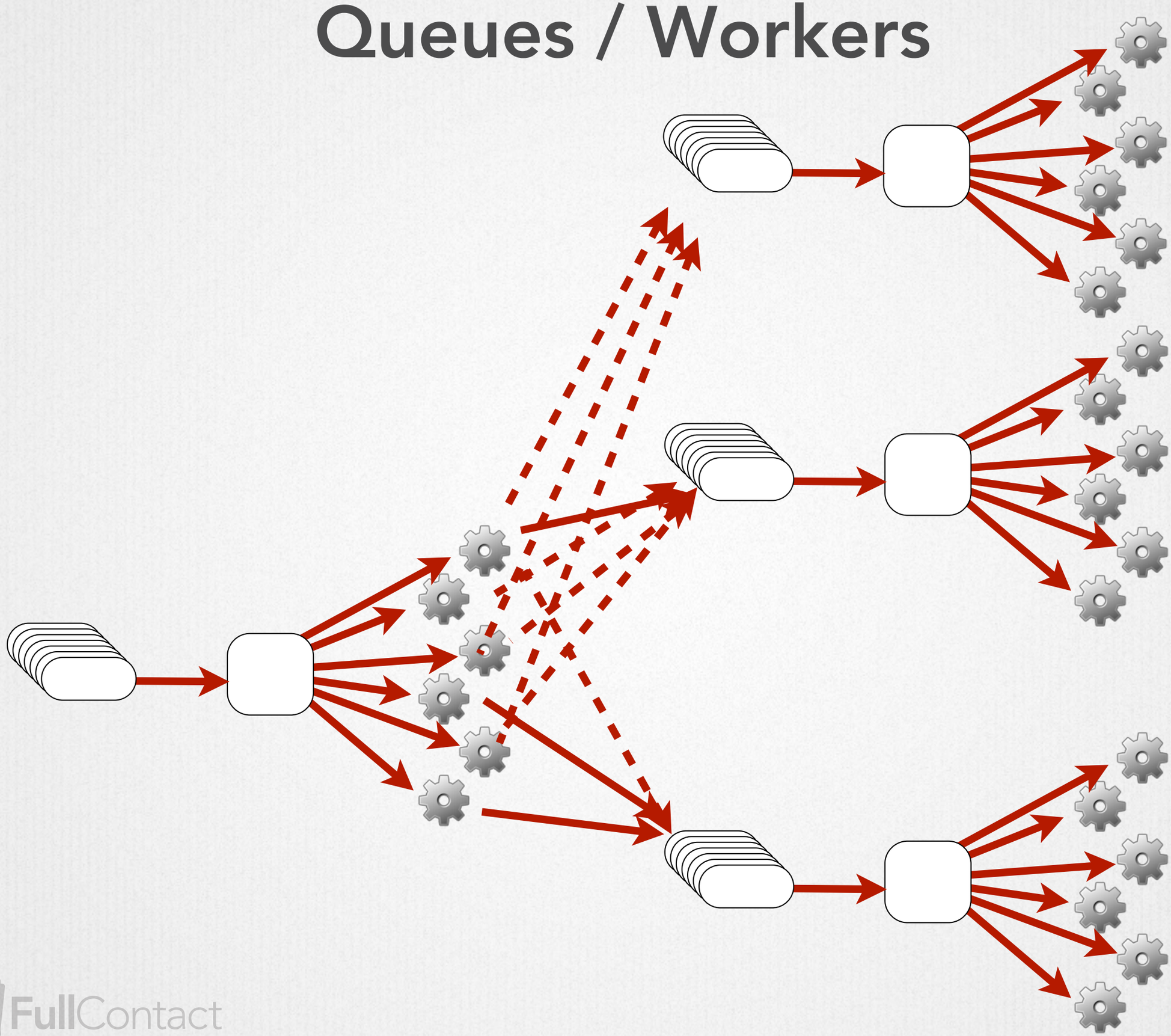


Queues / Workers



FullContact

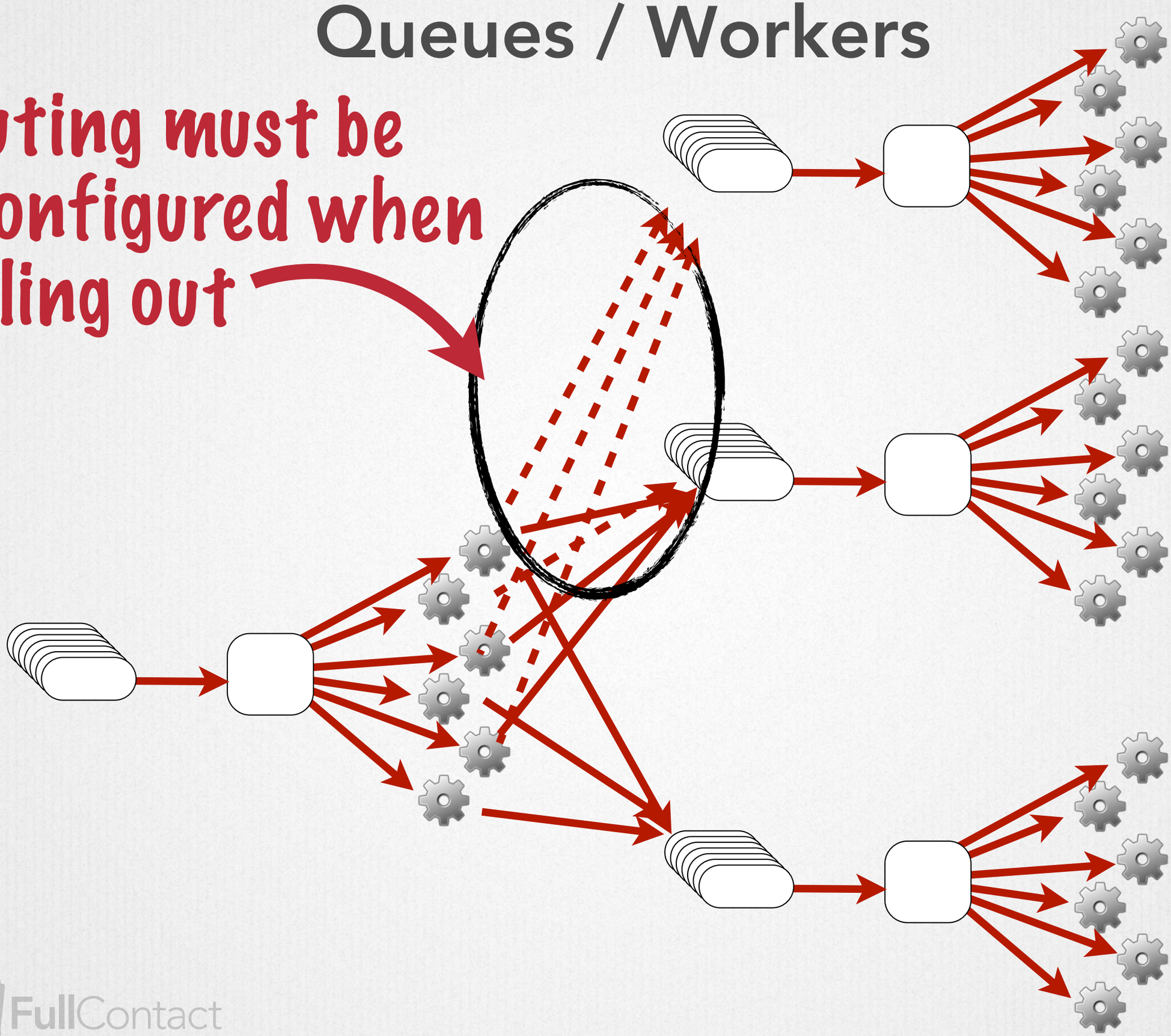
Queues / Workers



FullContact

Queues / Workers

Routing must be reconfigured when scaling out



FullContact

Storm

Storm

Distributed and fault-tolerant real-time computation

Storm

Distributed and fault-tolerant real-time computation

Storm

Distributed and fault-tolerant real-time computation

Storm

Distributed and fault-tolerant real-time computation

Key Concepts

FullContact

Tuples

Ordered list of elements

FullContact

Tuples

Ordered list of elements

```
("search-01384", "e:dan@fullcontact.com")
```

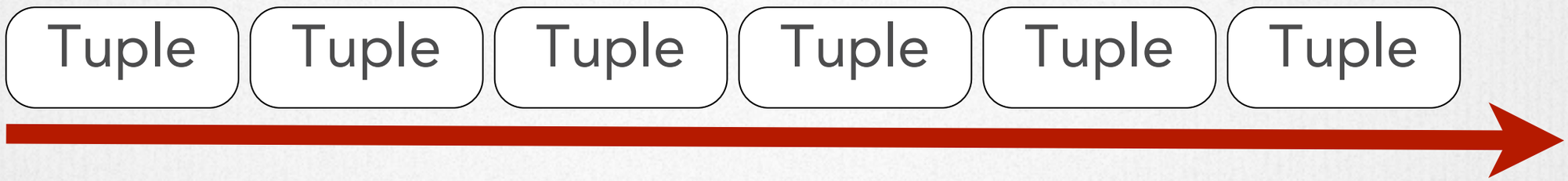
Streams

Unbounded sequence of tuples

FullContact

Streams

Unbounded sequence of tuples



FullContact

Spouts

Source of streams

FullContact

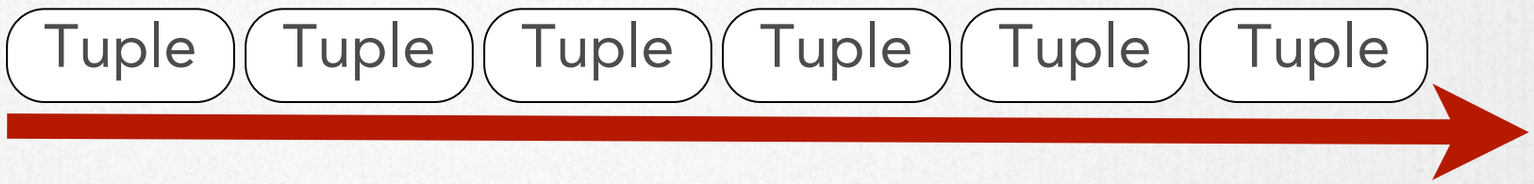
Spouts

Source of streams

FullContact

Spouts

Source of streams



FullContact

Spouts can talk with

some images from http://commons.wikimedia.org

Spouts can talk with

- Queues

some images from http://commons.wikimedia.org

Spouts can talk with

- Queues
- Web logs

some images from http://commons.wikimedia.org

Spouts can talk with

- Queues
- Web logs
- API calls

FullContact

some images from http://commons.wikimedia.org

Spouts can talk with

- Queues
- Web logs
- API calls
- Event data

FullContact

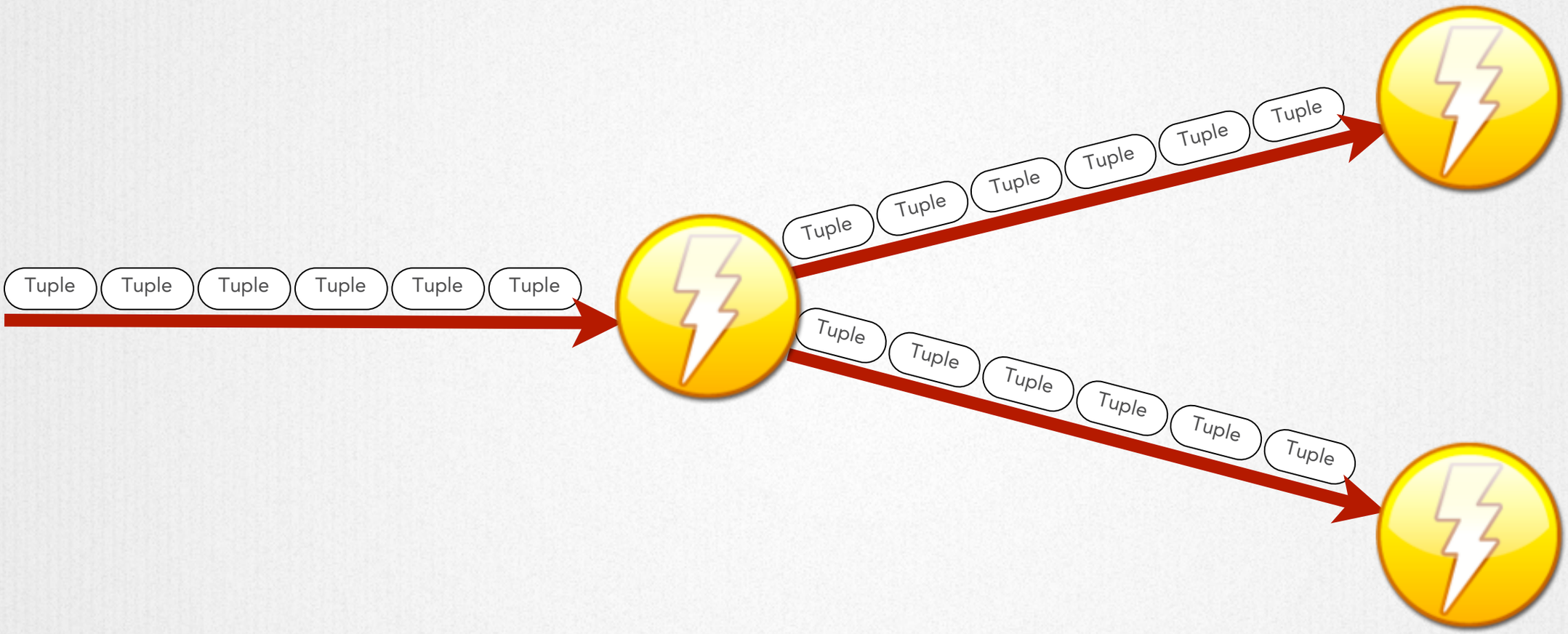
some images from http://commons.wikimedia.org

Bolts

Process tuples and create new streams

FullContact

Bolts



FullContact

some images from http://commons.wikimedia.org

Bolts

FullContact

some images from http://commons.wikimedia.org

Bolts

- Apply functions / transforms

FullContact

some images from http://commons.wikimedia.org

Bolts

- Apply functions / transforms
- Filter

FullContact

some images from http://commons.wikimedia.org

Bolts

- Apply functions / transforms
- Filter
- Aggregation

FullContact

some images from http://commons.wikimedia.org

Bolts

- Apply functions / transforms
- Filter
- Aggregation
- Streaming joins

FullContact

some images from http://commons.wikimedia.org

Bolts

- Apply functions / transforms
- Filter
- Aggregation
- Streaming joins
- Access DBs, APIs, etc...

FullContact

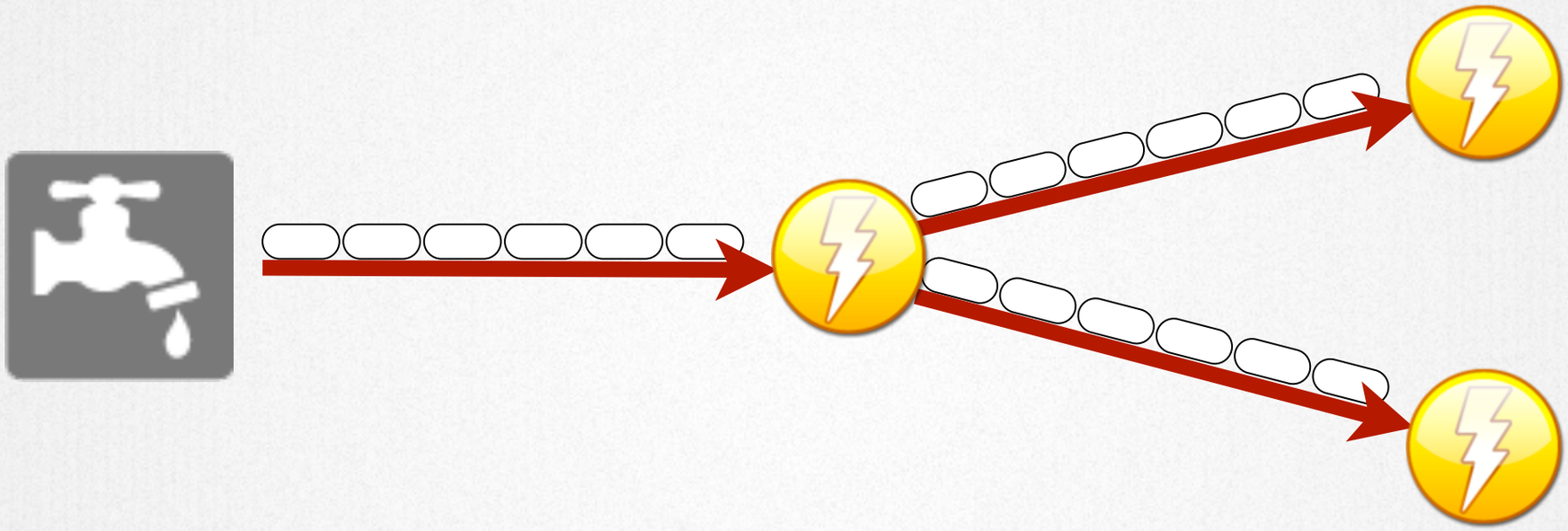
some images from http://commons.wikimedia.org

Topologies

A directed graph of Spouts and Bolts

FullContact

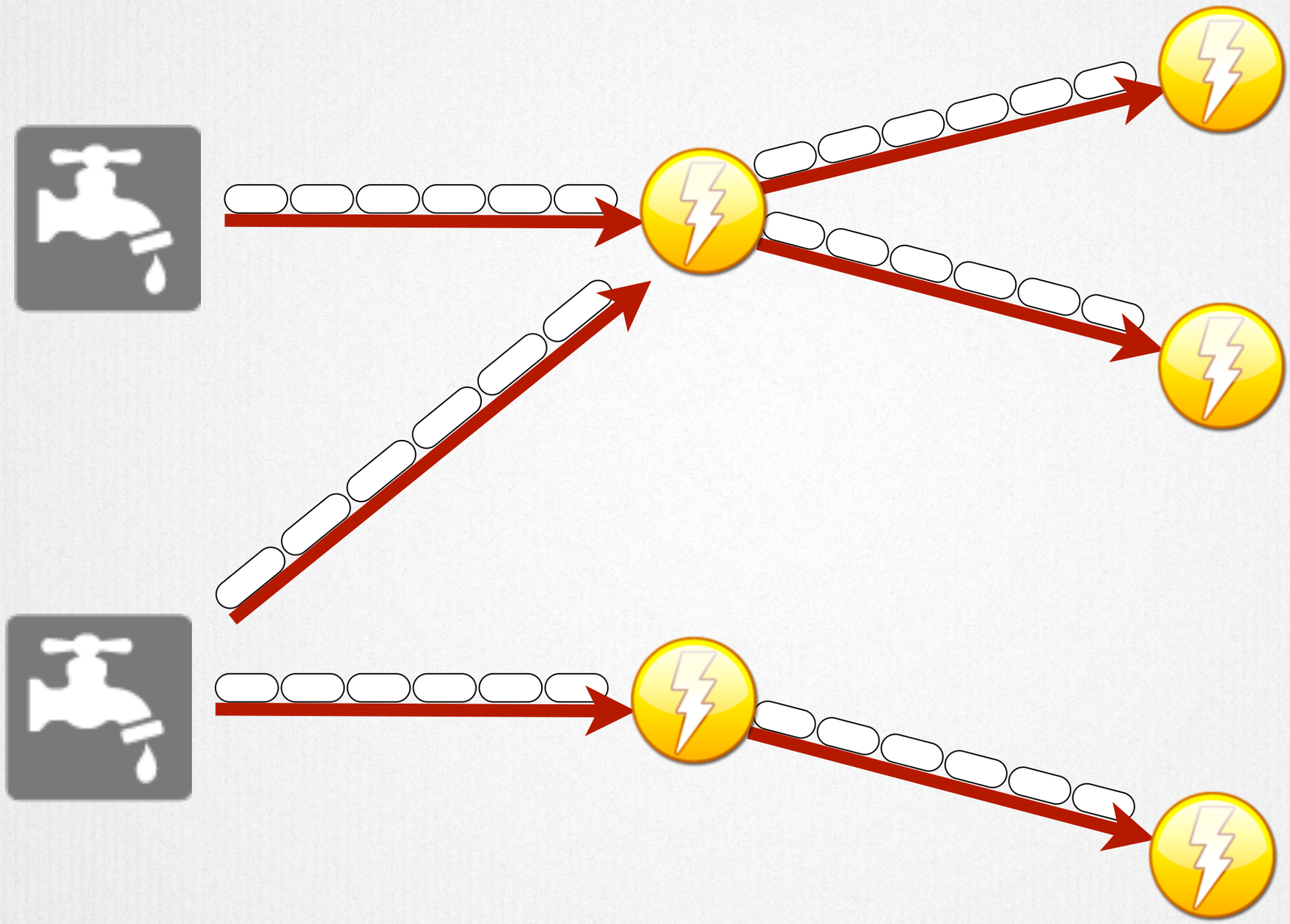
This is a Topology



FullContact

some images from http://commons.wikimedia.org

This is also a topology



FullContact

some images from http://commons.wikimedia.org

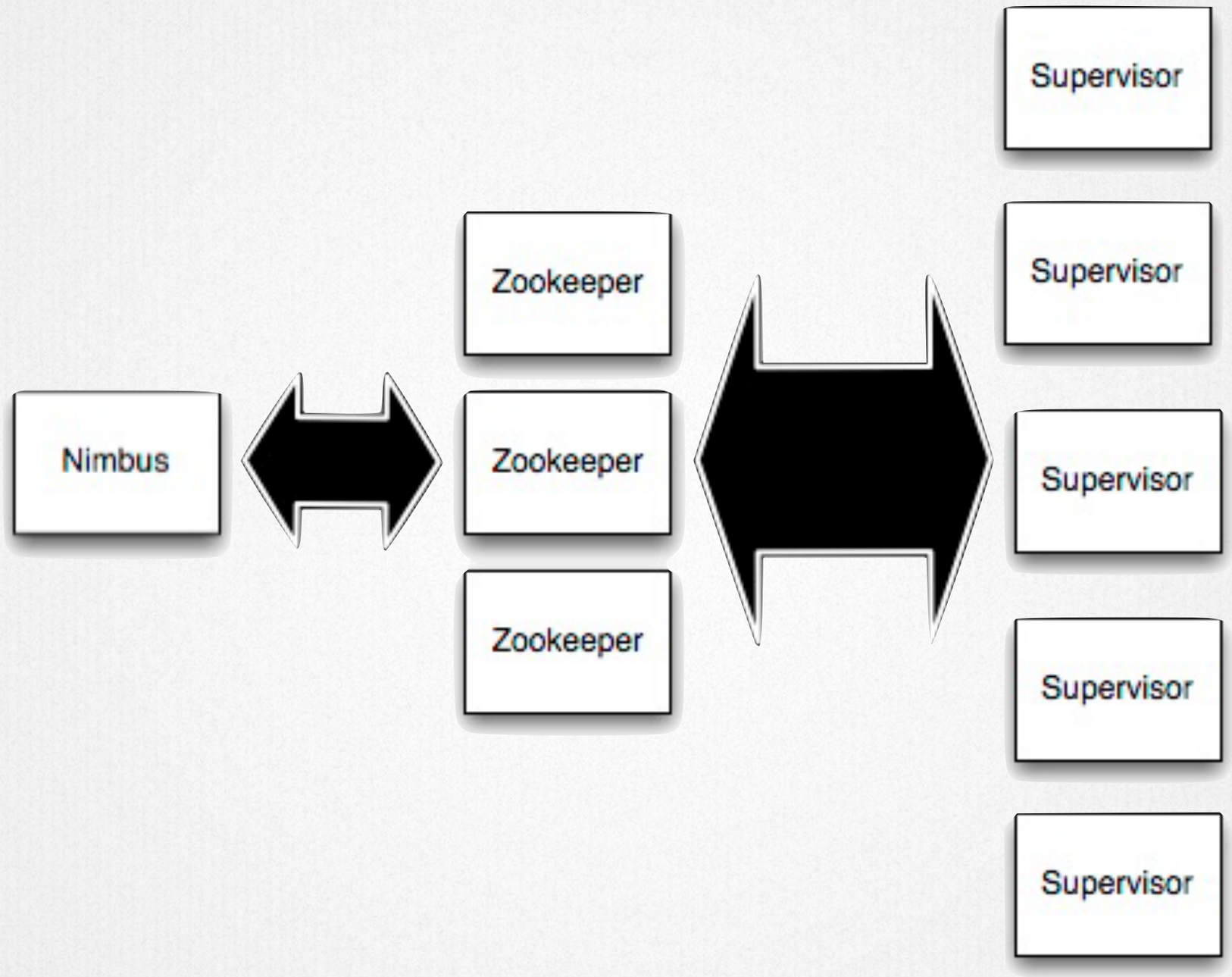
Tasks

Processes which execute Streams or Bolts

Running a Topology

```
$ storm jar my-code.jar com.example.MyTopology arg1 arg2
```

Storm Cluster

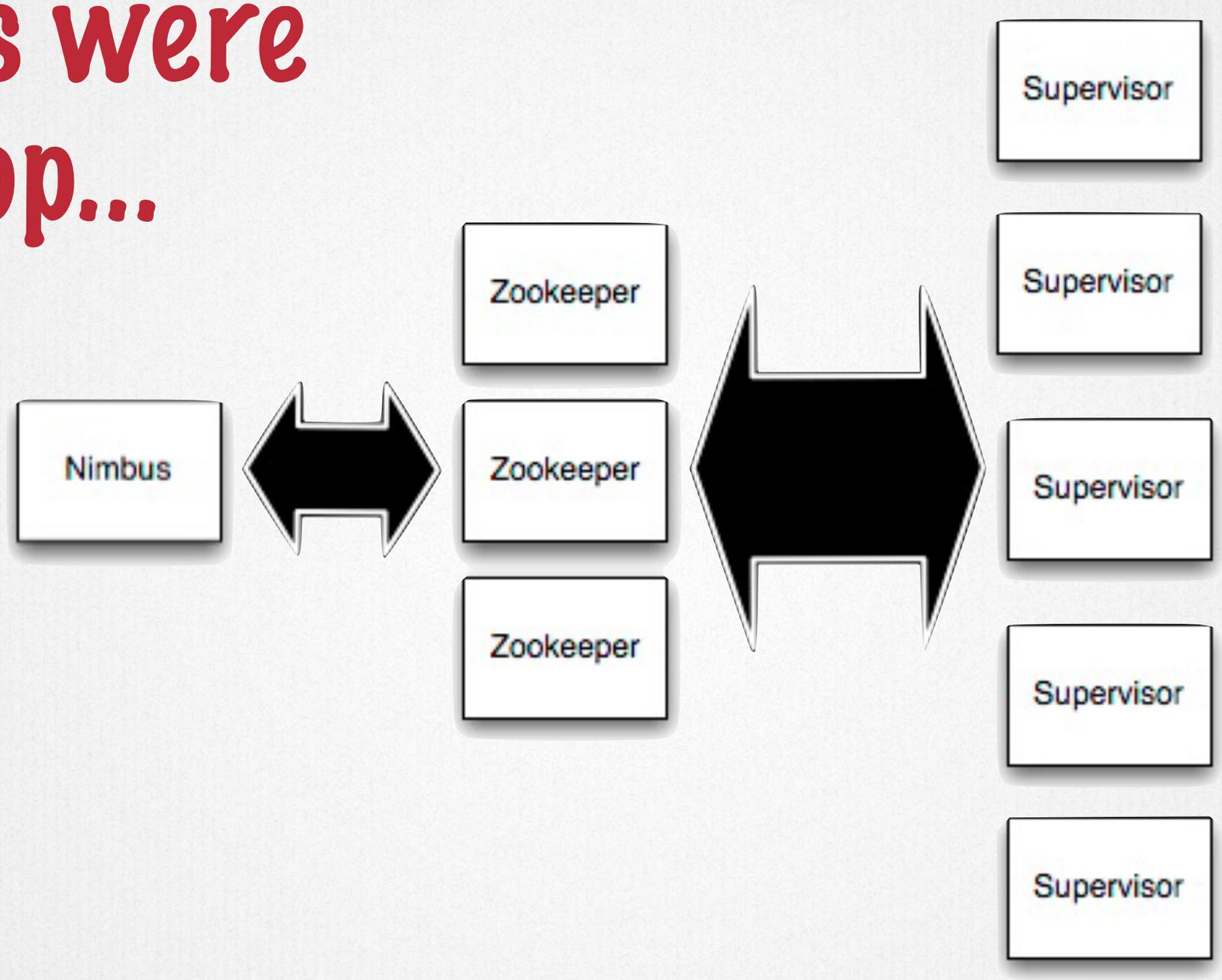


Nathan Marz

FullContact

Storm Cluster

If this were Hadoop...

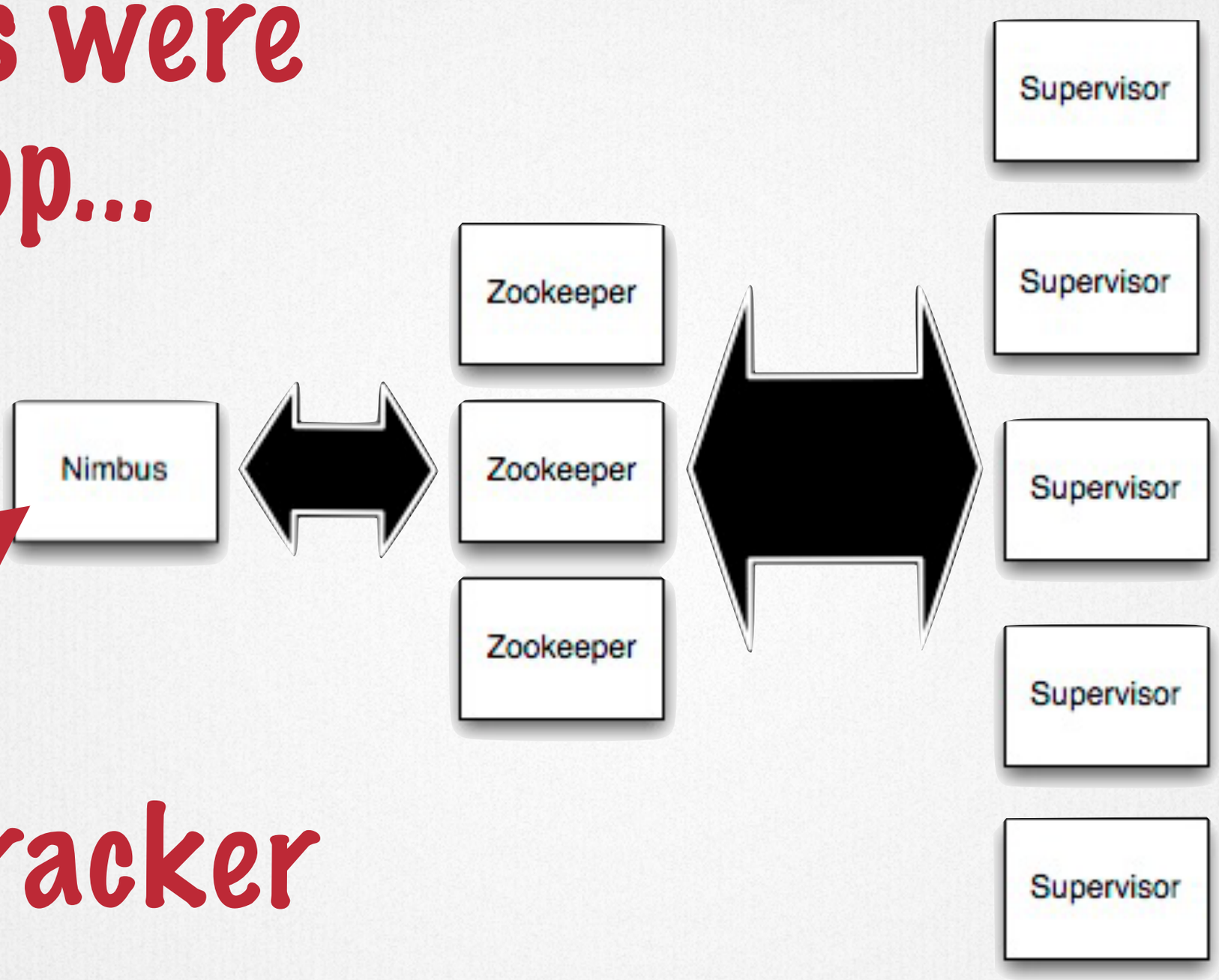


Nathan Marz

FullContact

Storm Cluster

If this were Hadoop...



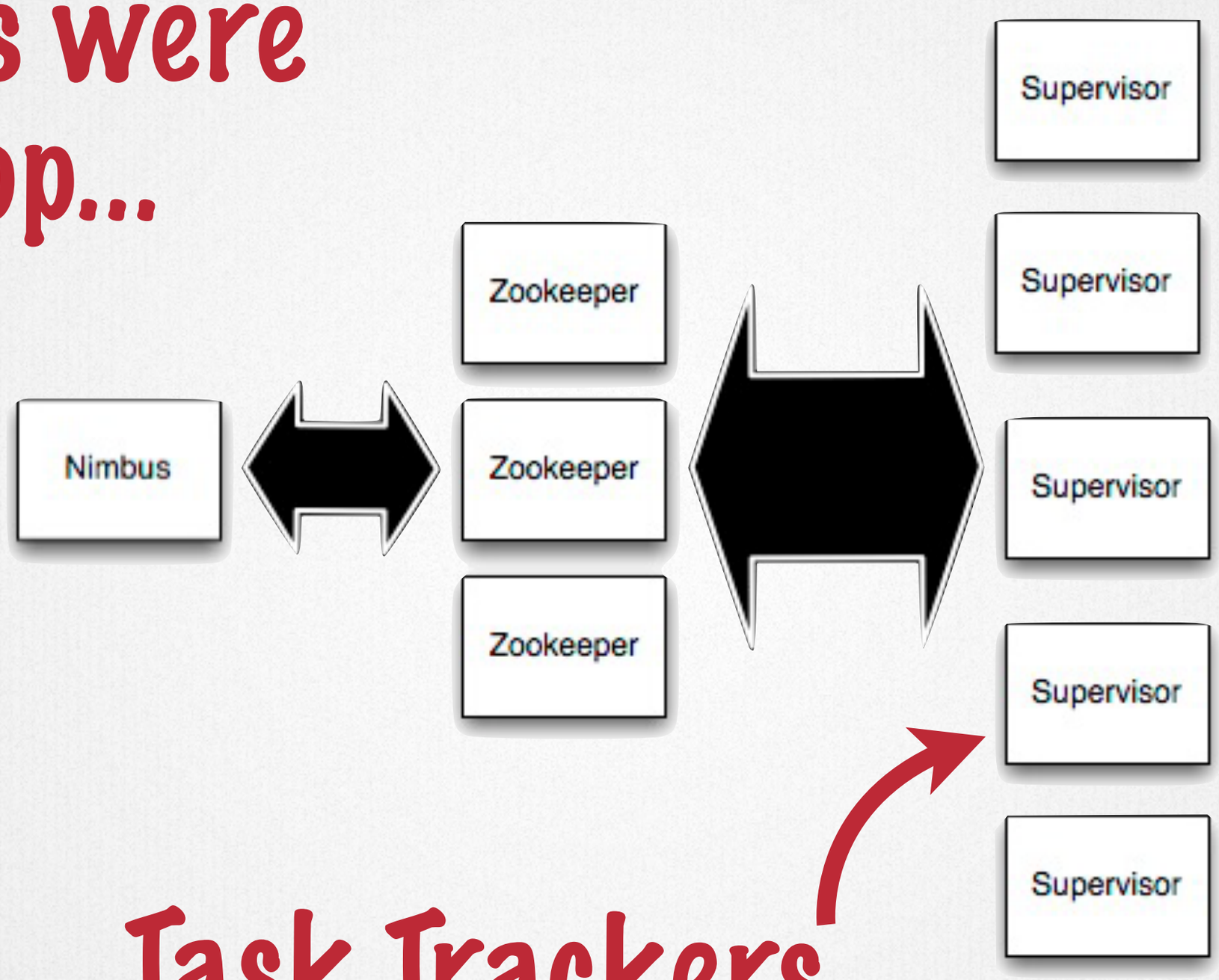
Job Tracker

Nathan Marz

FullContact

Storm Cluster

If this were Hadoop...



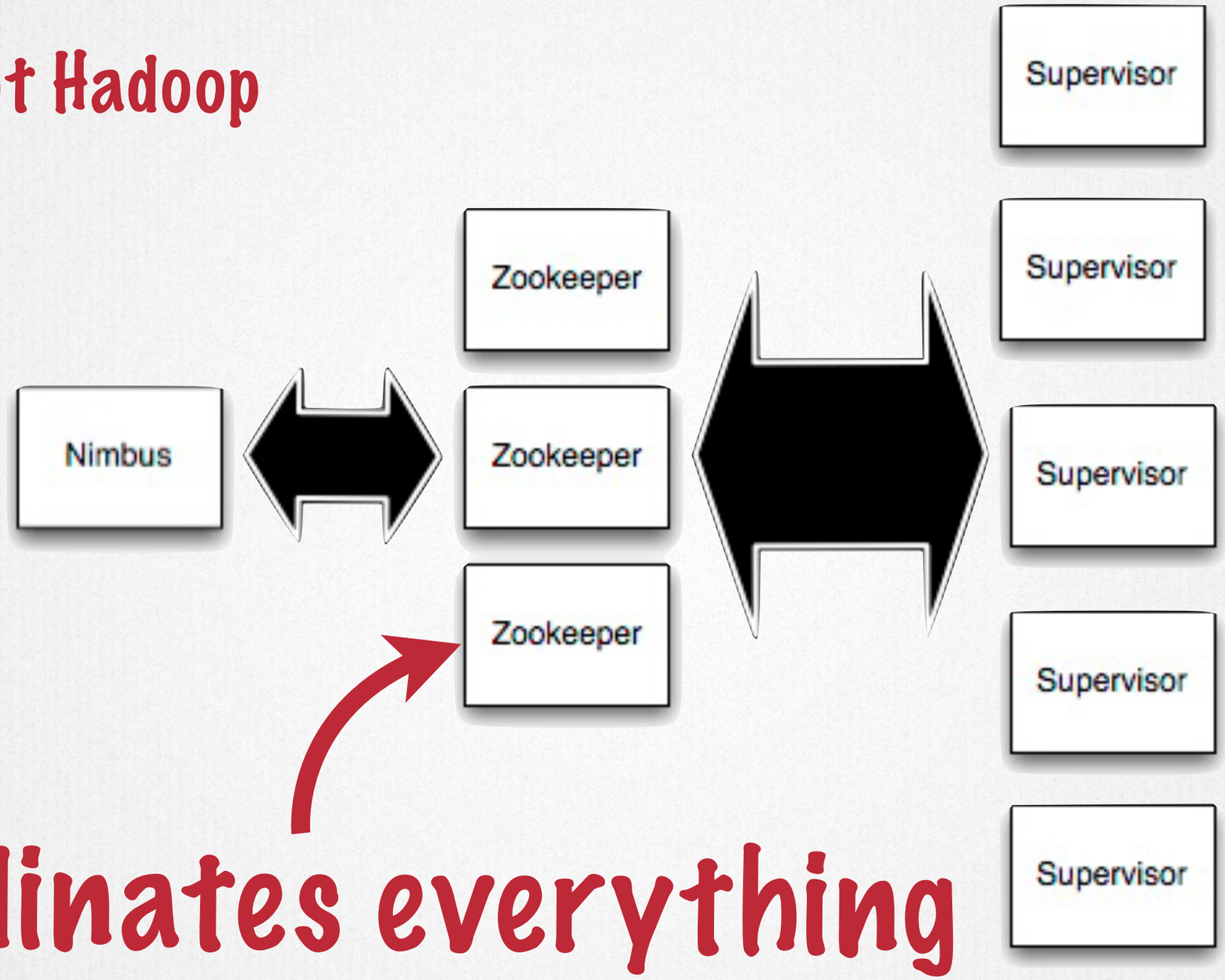
Task Trackers

Nathan Marz

FullContact

Storm Cluster

But it's not Hadoop



Coordinates everything

Nathan Marz

FullContact

Example:

Streaming Word Count

FullContact

Streaming Word Count

```
TopologyBuilder builder = new TopologyBuilder();

builder.setSpout("sentences", new RandomSentenceSpout(), 5);
builder.setBolt("split", new SplitSentence(), 8)
        .shuffleGrouping("sentences");
builder.setBolt("count", new WordCount(), 12)
        .fieldsGrouping("split", new Fields("word"));
```

Streaming Word Count

```
TopologyBuilder builder = new TopologyBuilder();

builder.setSpout("sentences", new RandomSentenceSpout(), 5);
builder.setBolt("split", new SplitSentence(), 8)
        .shuffleGrouping("sentences");
builder.setBolt("count", new WordCount(), 12)
        .fieldsGrouping("split", new Fields("word"));
```

Streaming Word Count

```
public static class SplitSentence extends ShellBolt implements IRichBolt {

    public SplitSentence() {
        super("python", "splitsentence.py");
    }

    @Override
    public void declareOutputFields(OutputFieldsDeclarer declarer) {
        declarer.declare(new Fields("word"));
    }

    @Override
    public Map<String, Object> getComponentConfiguration() {
        return null;
    }
}
```

SplitSentence.java

Streaming Word Count

```java
public static class SplitSentence extends ShellBolt implements IRichBolt {

    public SplitSentence() {
        super("python", "splitsentence.py");
    }

    @Override
    public void declareOutputFields(OutputFieldsDeclarer declarer) {
        declarer.declare(new Fields("word"));
    }

    @Over
    publi
        r
    }
}
```

SplitSentence.java

```python
import storm

class SplitSentenceBolt(storm.BasicBolt):
    def process(self, tup):
        words = tup.values[0].split(" ")
        for word in words:
            storm.emit([word])
```

splitsentence.py

Streaming Word Count

```
public static class SplitSentence extends ShellBolt implements IRichBolt {

    public SplitSentence() {
        super("python", "splitsentence.py");
    }

    @Override
    public void declareOutputFields(OutputFieldsDeclarer declarer) {
        declarer.declare(new Fields("word"));
    }

    @Override
    public Map<String, Object> getComponentConfiguration() {
        return null;
    }
}
```

SplitSentence.java

Streaming Word Count

```java
TopologyBuilder builder = new TopologyBuilder();

builder.setSpout("sentences", new RandomSentenceSpout(), 5);
builder.setBolt("split", new SplitSentence(), 8)
        .shuffleGrouping("sentences");
builder.setBolt("count", new WordCount(), 12)
        .fieldsGrouping("split", new Fields("word"));
```

java

Streaming Word Count

```
public static class WordCount extends BaseBasicBolt {
    Map<String, Integer> counts = new HashMap<String, Integer>();

    @Override
    public void execute(Tuple tuple, BasicOutputCollector collector) {
        String word = tuple.getString(0);
        Integer count = counts.get(word);
        if(count==null) count = 0;
        count++;
        counts.put(word, count);
        collector.emit(new Values(word, count));
    }

    @Override
    public void declareOutputFields(OutputFieldsDeclarer declarer) {
        declarer.declare(new Fields("word", "count"));
    }
}
```

WordCount.java

Streaming Word Count

```java
TopologyBuilder builder = new TopologyBuilder();

builder.setSpout("sentences", new RandomSentenceSpout(), 5);
builder.setBolt("split", new SplitSentence(), 8)
        .shuffleGrouping("sentences");
builder.setBolt("count", new WordCount(), 12)
        .fieldsGrouping("split", new Fields("word"));
```

java

Groupings control how tuples are routed

Shuffle grouping

Tuples are randomly distributed across all of the tasks running the bolt

Fields grouping

Groups tuples by specific named fields and routes them to the same task

Analogous to Hadoop's partitioning behavior

Fields grouping

Groups tuples by specific named fields and routes them to the same task

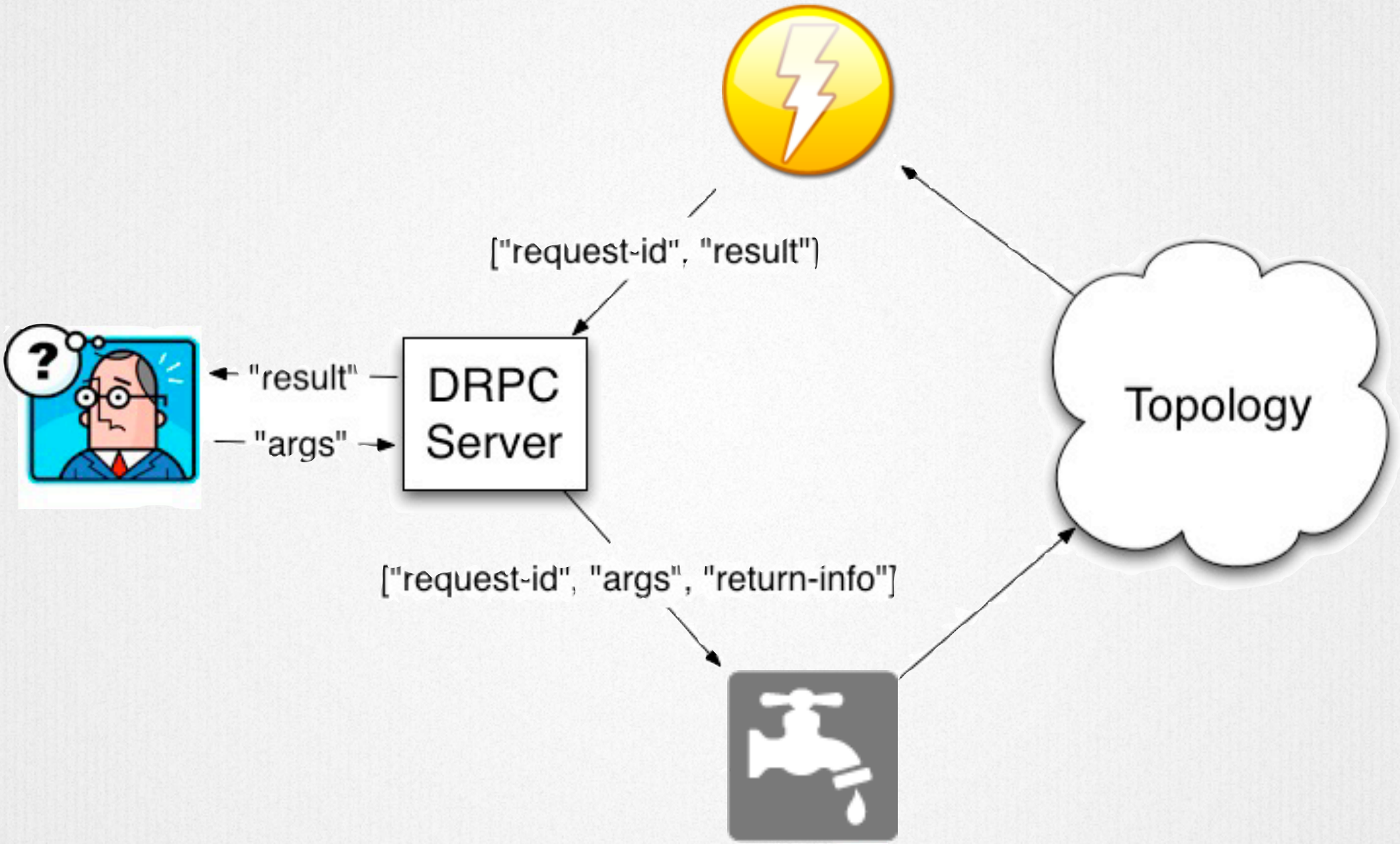
Distributed RPC

Before Distributed RPC,
time-sensitive queries relied on a
pre-computed index

What if you didn’t need an index?

FullContact

Distributed RPC



FullContact

Try it out!

Huge thanks to Nathan Marz - @nathanmarz

http://github.com/nathanmarz/storm

https://github.com/nathanmarz/storm-starter

@stormprocessor

Questions?

dan@fullcontact.com