Applications Scenarios for Large Scale Stream Processing

JKU
JOHANNES KEPLER
UNIVERSITY LINZ

Matthias Steinbauer
Institute of Telekooperation

Data becomes so

big

in terms of

size/volume

such that it becomes

awkward to handle


A big
variety
of data
needs to be
processed

data arrives at
very high
velocities

“We leave
traces
of ourselves
wherever we go,
on whatever we touch.”

Lewis Thomas

twitter.com handles **500** million tweets each day

http://www.internetlivestats.com/twitter-statistics/

Bluetooth

Location

e-Mail

Text Messages

Acceleration

Phone Calls

Compass

Other high velocity data-sources

- Log files: Devices, databases, web-servers; any technology creates log file

- IT Devices: firewalls, printers, every device outputs valuable data

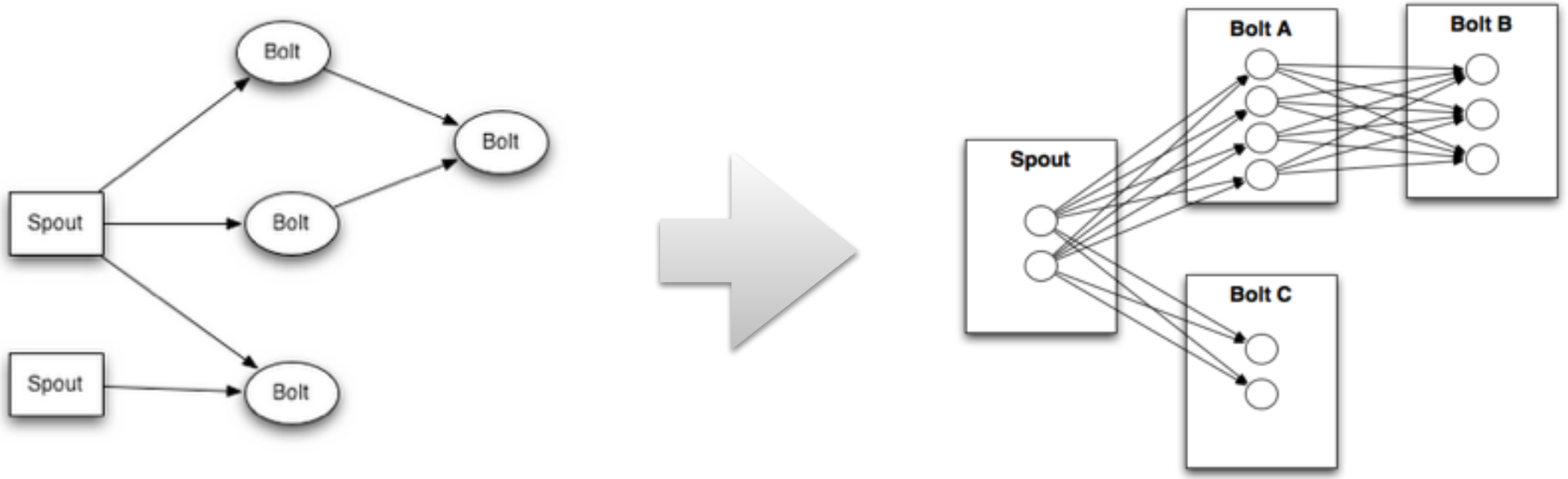
- User devices: Mobile phones create a plethora of data

- Online gaming: Massive Multiplayer Online Gaming systems create massive amounts of data

- SaaS applications can produce large amounts of streaming data if they grow in size

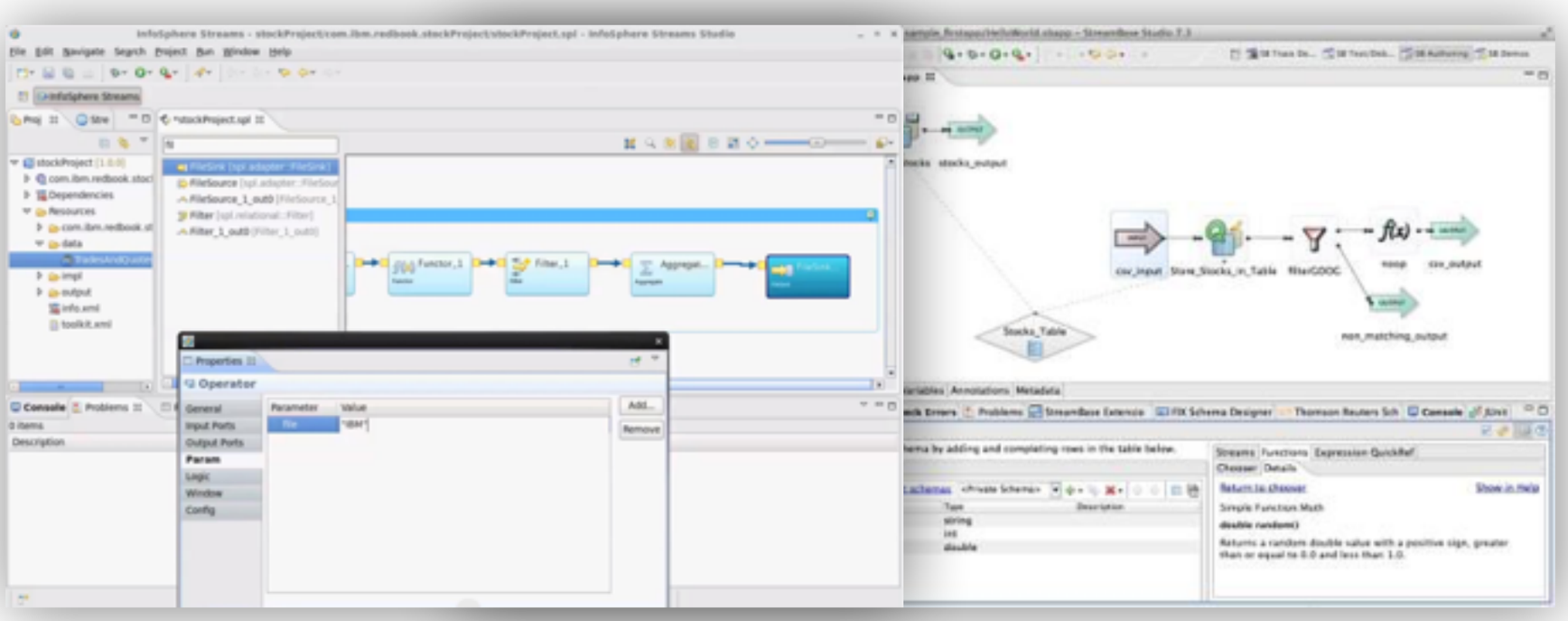
Big Data Stream Processors

- Developers define a “topology” for the data-flow
- The stream processor finds an intelligent placement for each component of the data-flow over real compute nodes



Implementations

- Apache Storm, Apache Samza, Apache Spark (although more general), Amazon Kinesis, IBM InfoSphere Streams, TIBICO StreamBase, etc.



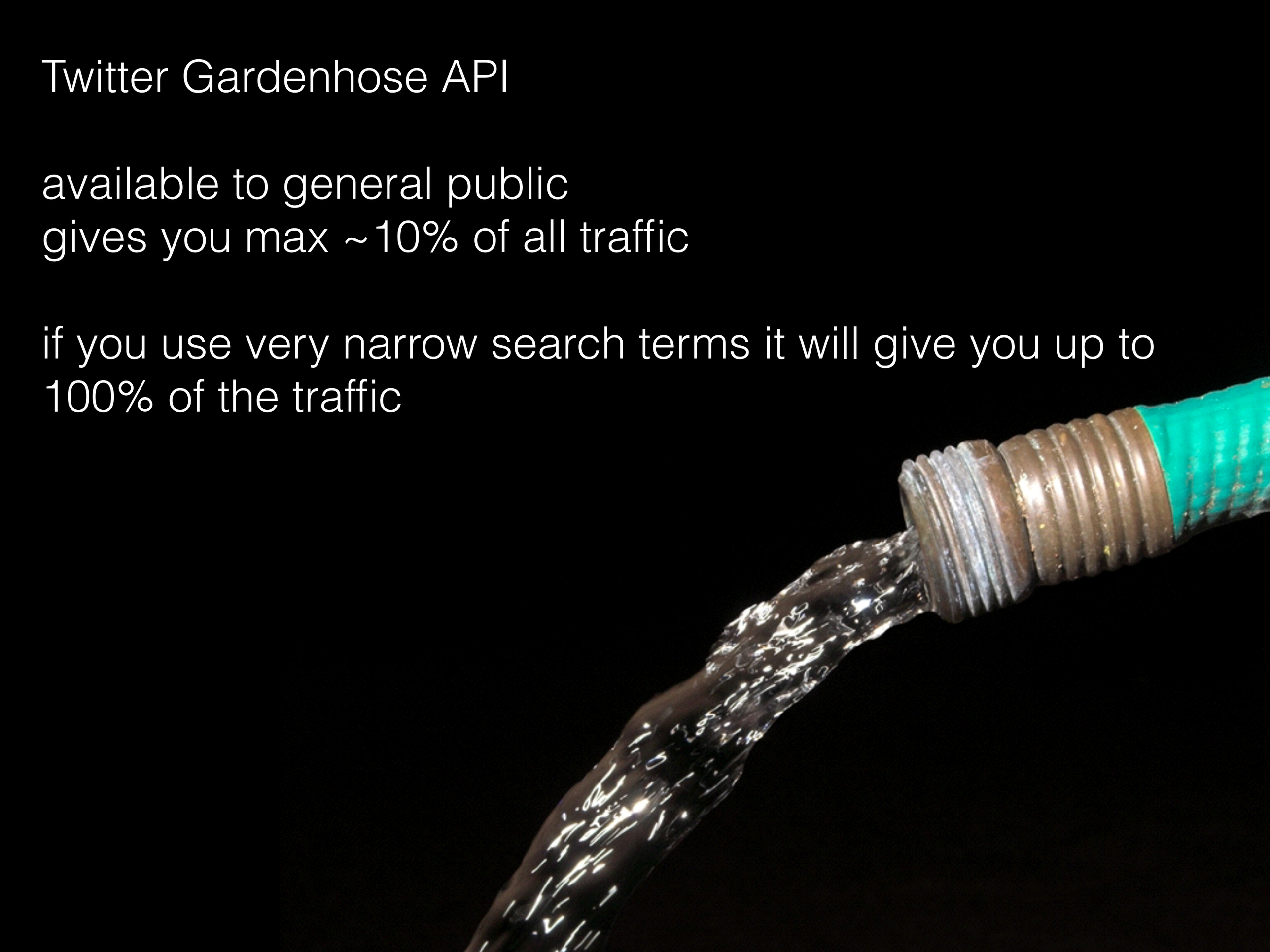
Example: Big Data from Social Media

- Data is the most important asset of social media companies such as Facebook, Twitter, LinkedIn etc.
- Thus data access is usually limited or costly
- Clever filtering will reveal interesting insight

Twitter Gardenhose API

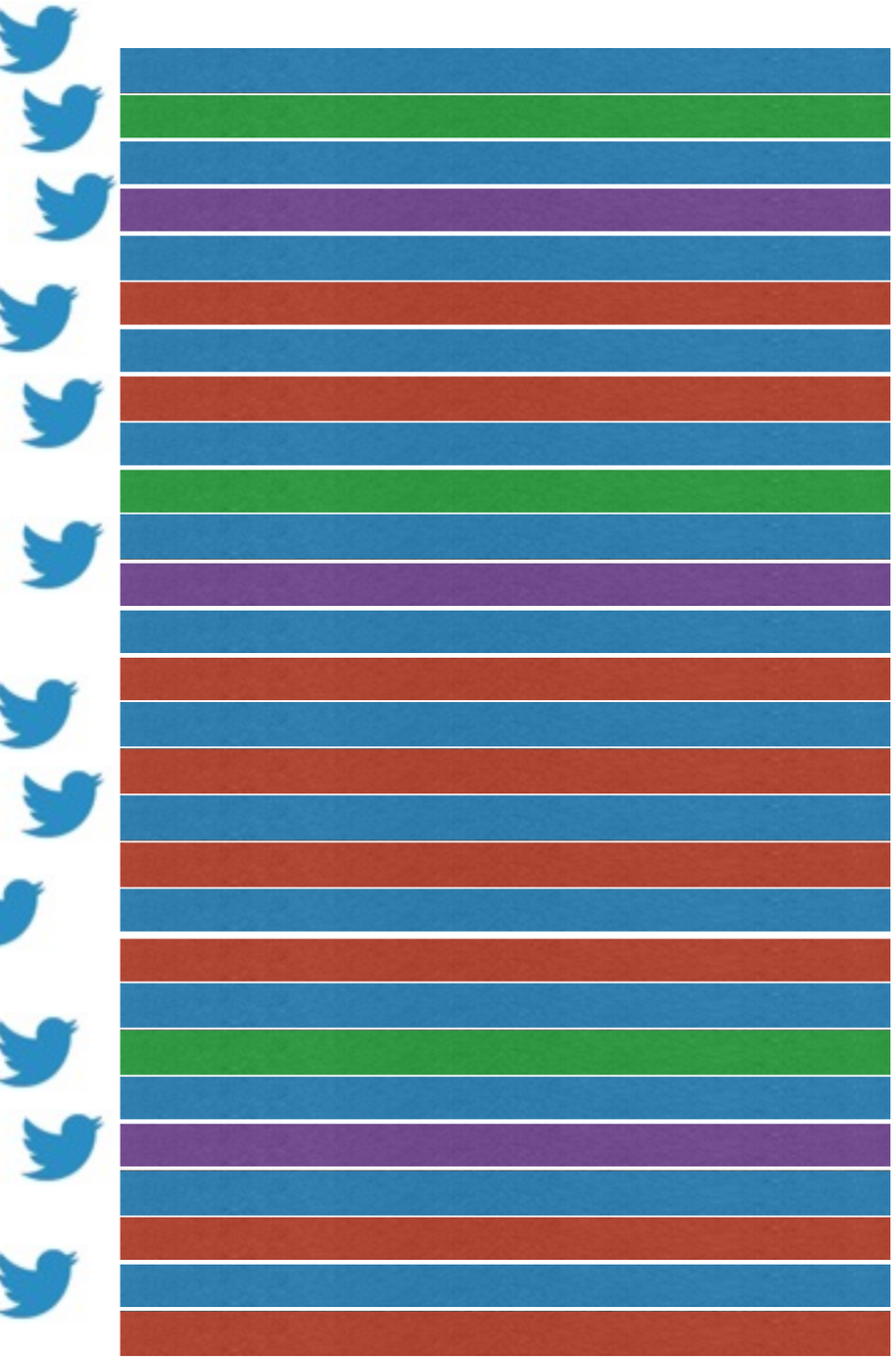
available to general public
gives you max ~10% of all traffic

if you use very narrow search terms it will give you up to 100% of the traffic

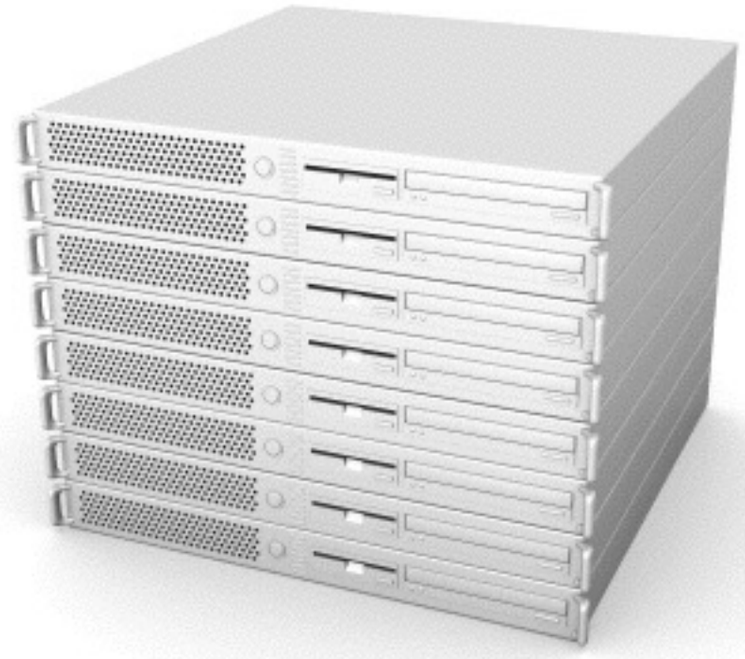


Firehose access:
100% of all Tweets in near real-time
(if your network can handle it)

What are the top 10 hashtags?



Compute top 10?



What are the top 10 hashtags?

Local Top 10

Global Top 10

What are the top 10 hashtags?

Top 10 grouped by tag

Aggregate

Real time social media sentiment analysis

Streams data directly from Twitter

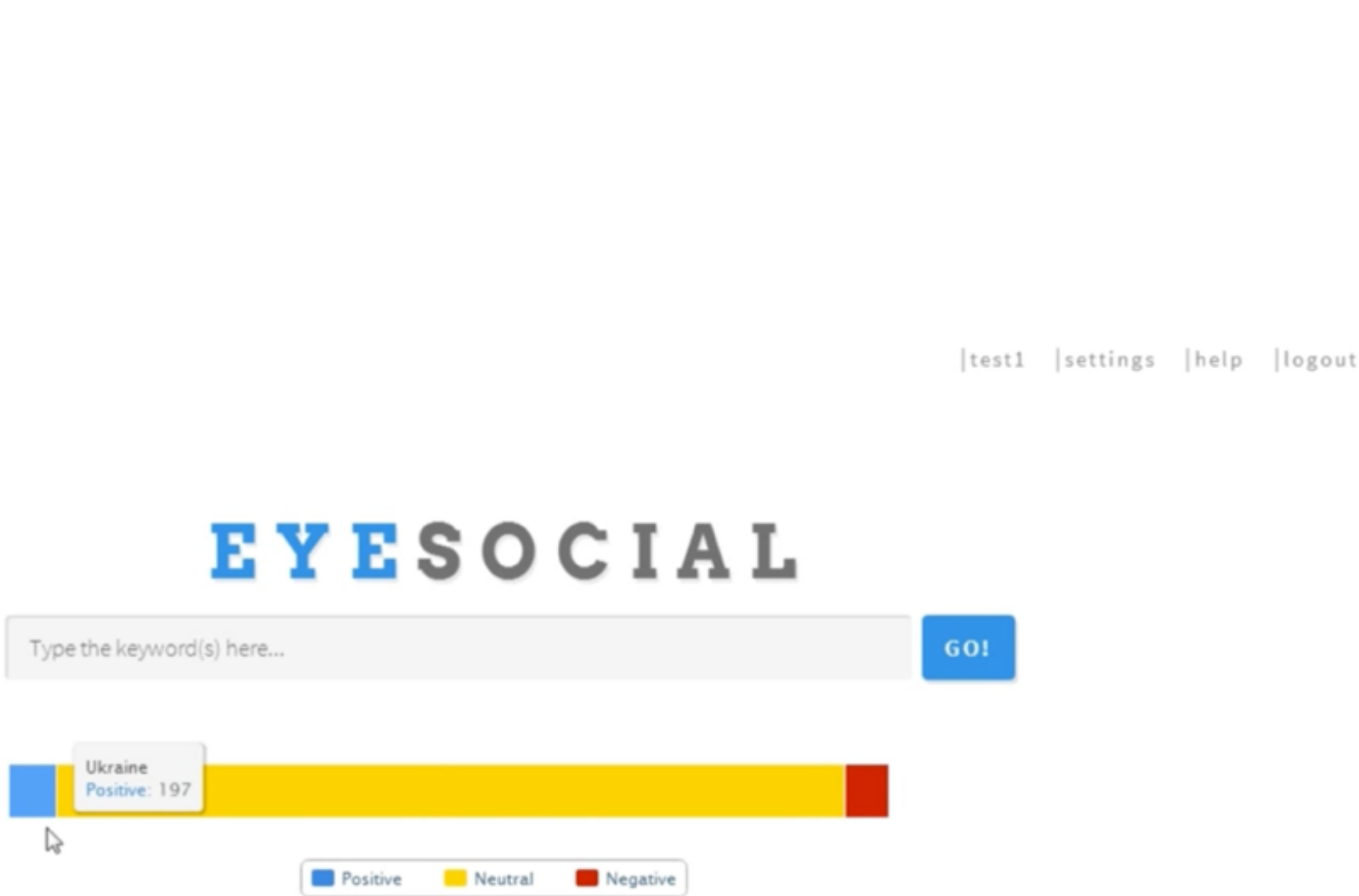
EYESOCIAL

Ukraine

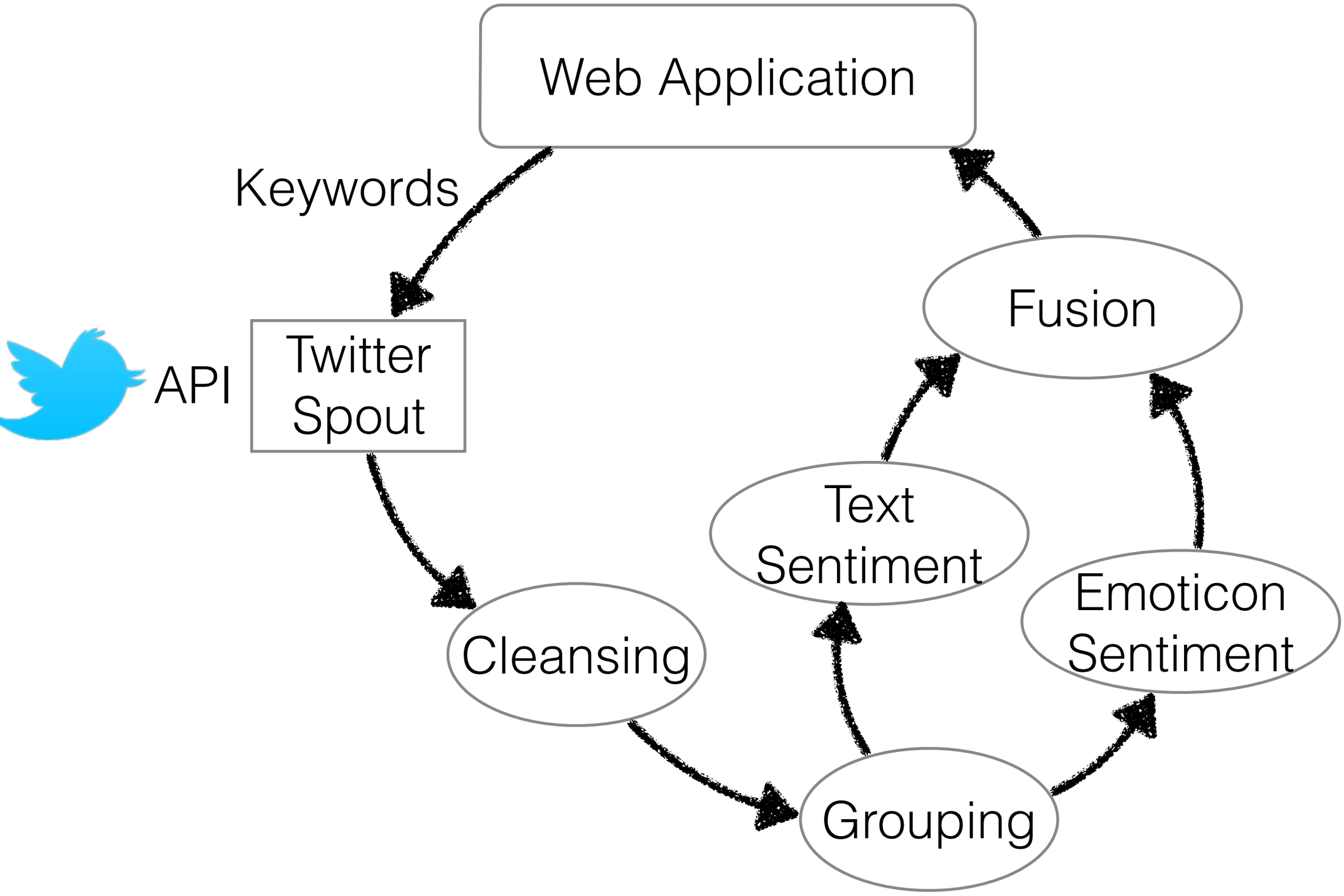
GO!

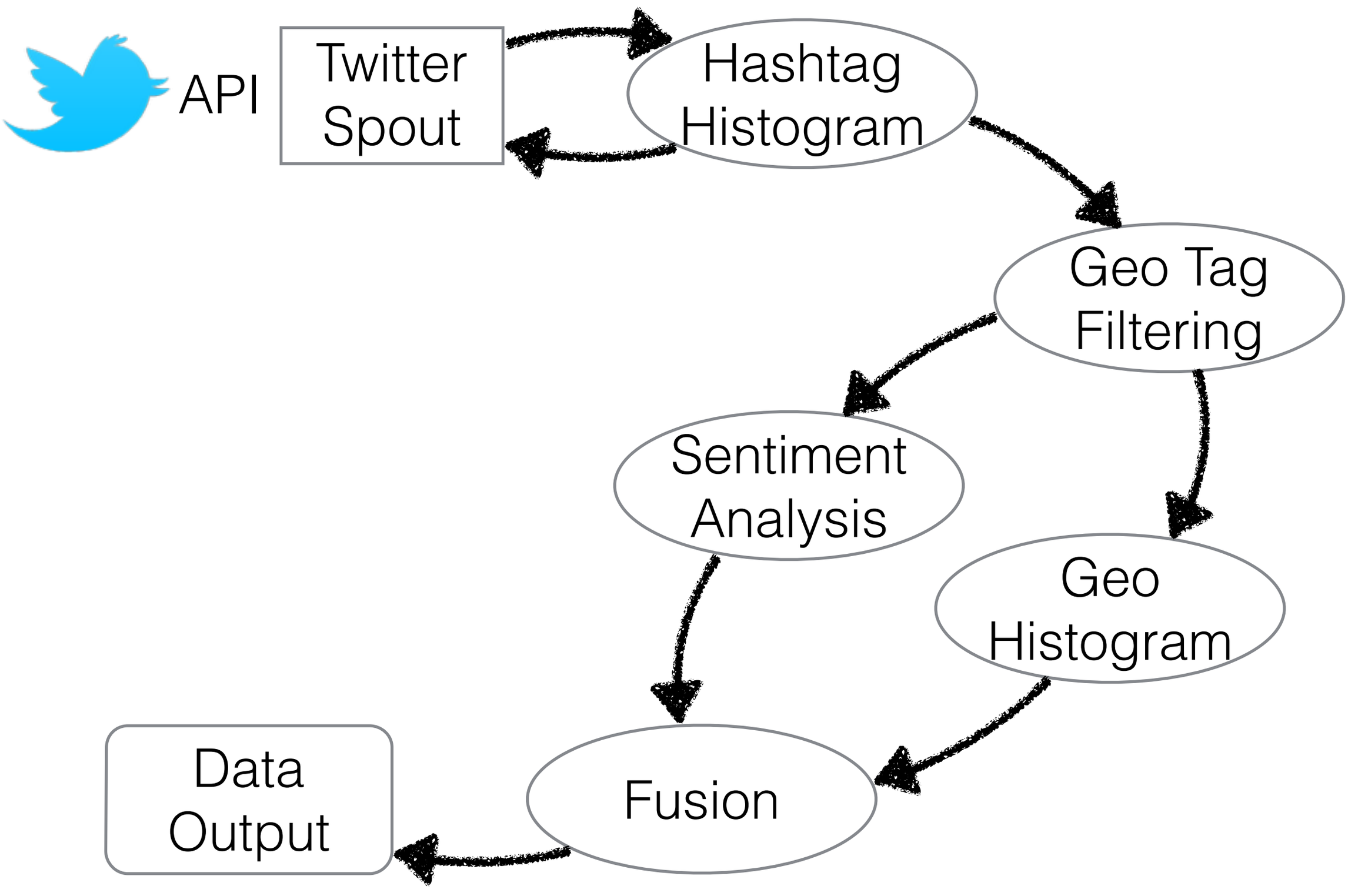
Total number of tweets processed in backend: 173

Positive Neutral Negative



https://www.youtube.com/watch?v=cgS5QMImVdU


Web Application
Keywords
API
Twitter
Spout
Cleansing
Grouping
Text
Sentiment
Fusion
Emoticon
Sentiment


API
Twitter Spout
Hashtag Histogram
Geo Tag Filtering
Sentiment Analysis
Geo Histogram
Fusion
Data Output

Department of Telecooperation

Analysis on semantically structured data at the scale of smart cities

ftw Creating Communication Technologies

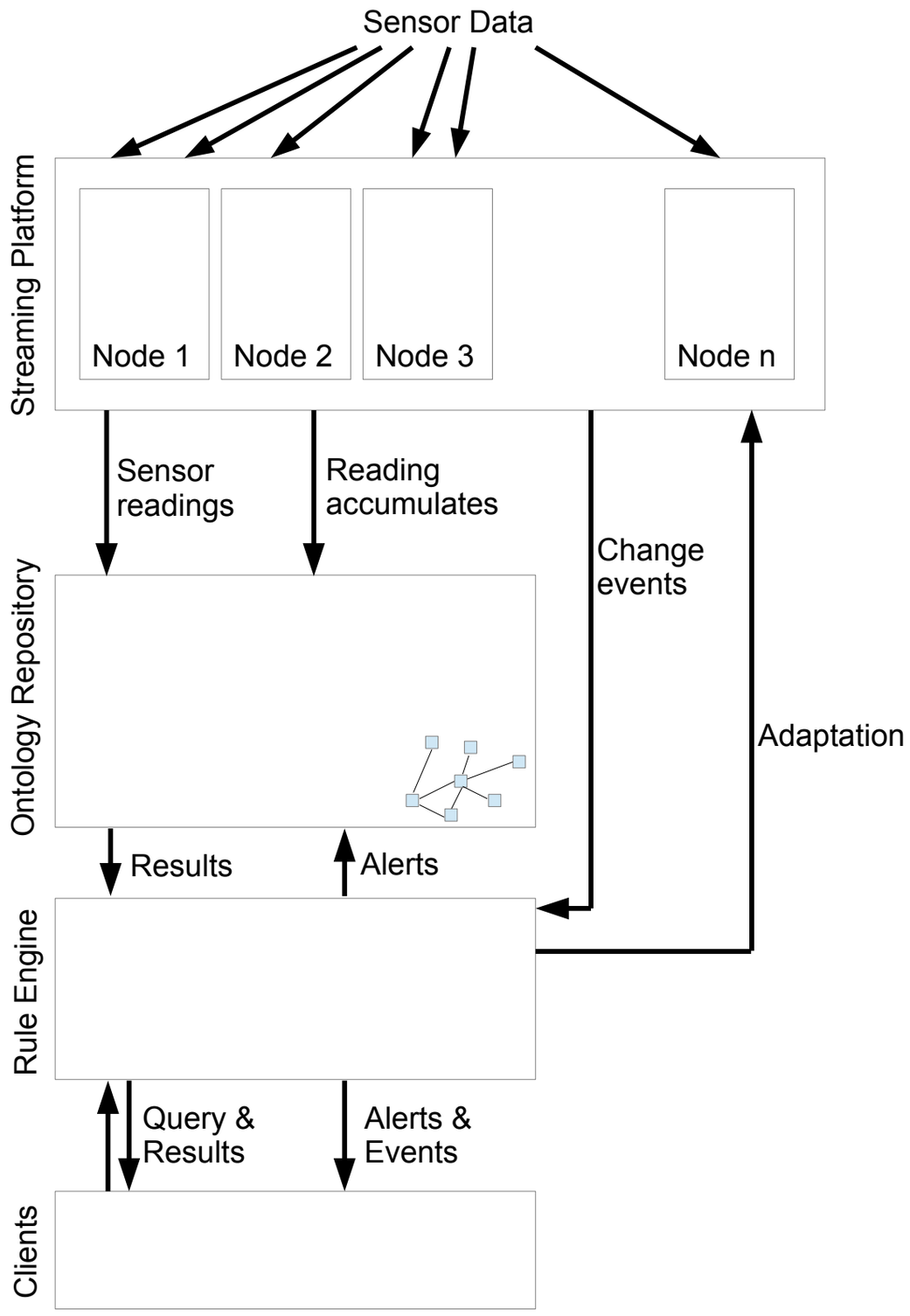
gabriele.kotsis@jku.ac.at

Privatperson: Ontologien erlauben uns Regeln die für ein Haus geleten einfach auf ein anderes anzuwenden

Energieanbieter: Verstehen das Nutzungsverhalten besser und können Energie effizienter bereitstellen
(Privatpersonen können zum Anbieter werden)

Datenmenge, Geschwindigkeit und Heterogenität → Big Data Problem

Große Datenmenge kann allerdings auf herkömmlichen Ontologie basierten Systemen nicht abgebildet werden



ftw Creating Communication Technologies

gabriele.kotsis@jku.ac.at

Logistics processes generate ever more data

- GPS trajectories
- package handling

allows for better:

- congestion handling
- user interaction
- planning

Industry 4.0

An industry where every productive piece in the line carries some intelligence and sensing capabilities

Some code anybody?

```
public class TestSpout extends BaseRichSpout {
    private static final long serialVersionUID = 2592201664248511961L;
    private SpoutOutputCollector collector;
    private Random rand;
    public void open(Map conf, TopologyContext context,
    SpoutOutputCollector collector) {
        this.collector = collector;
        this. rand = new Random();
    }
    public void declareOutputFields(OutputFieldsDeclarer declarer) {
        declarer.declare(new Fields("sentence"));
    }
    public void nextTuple() {
        Utils.sleep(100);
        String[] sentences = new String[] {
            "the cow jumped over the moon", "an apple a day keeps the doctor away",
            "four score and seven years ago", "snow white and the seven dwarfs",
            "i am at two with nature"
        };
        String sentence = sentences[rand.nextInt(sentences.length)];
        collector.emit(new Values(sentence));
    }

    @Override public void ack(Object id) { }

    @Override public void fail(Object id) { }
}
```

```
public class SplitSentenceBolt extends BaseBasicBolt {

    private static final long serialVersionUID = 8434746936407196175L;

    public void execute(Tuple input, BasicOutputCollector collector) {
        String sentence = input.getStringByField("sentence");
          // TODO too simple but just for an example
        String[] words = sentence.split(" ");
        for(String w : words) {
            collector.emit(new Values(w));
        }
    }

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

}
```

```
public class WordCountBolt extends BaseBasicBolt {

    private static final long serialVersionUID = 5570507244464043766L;

    private HashMap<String, Integer> counts = new HashMap<String, Integer>();

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

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

}
```

```
public class WriteLogFilesBolt extends BaseBasicBolt {

    private static final long serialVersionUID = -8773471129312629498L;

    public void execute(Tuple input, BasicOutputCollector collector) {
        String word = input.getStringByField("word");
        Integer count = input.getIntegerByField("count");
        File file = new File("/tmp/word-count.log");
        try {
            FileWriter bw = new FileWriter(file, true);
            bw.append(word); bw.append(": ");
            bw.append(count.toString()); bw.append("\n"); bw.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void declareOutputFields(OutputFieldsDeclarer declarer) {
        // Do not declare output we just save what we receive
    }

}
```

```
public class App {
    public static void main(String[] args)
    throws InterruptedException, AlreadyAliveException, InvalidTopologyException {
        TopologyBuilder topoBuilder = new TopologyBuilder();
        topoBuilder.setSpout("spout", new TestSpout(), 1);
        topoBuilder.setBolt("split",
      new SplitSentenceBolt(), 10).shuffleGrouping("spout");
        topoBuilder.setBolt("count",
      new WordCountBolt(), 10).fieldsGrouping("split", new Fields("word"));
        topoBuilder.setBolt("logs",
      new WriteToServerBolt(), 1).shuffleGrouping("count");

        Config conf = new Config();
        conf.setDebug(true);

        if (args!=null && args.length > 0) {
            conf.setNumWorkers(9);
            StormSubmitter.submitTopology(args[0], conf, topoBuilder.createTopology());
        }else{
            conf.setMaxTaskParallelism(3);
            LocalCluster cluster = new LocalCluster();
            cluster.submitTopology("word-count", conf, topoBuilder.createTopology());
            Thread.sleep(10000);
            cluster.shutdown();
        }
    }
}
```

JKU
JOHANNES KEPLER
UNIVERSITY LINZ

Code at
https://github.com/jkutk/

Matthias Steinbauer
matthias.steinbauer@jku.at
@climbinggeek