

BDI

BIG DATA INSTITUTE

Processing Data of Any Size with Apache Beam

About Big Data Institute

Mentoring, training, and high-level consulting company focused on Big Data, NoSQL and The Cloud

- Founded in 2008

We help make companies successful with Big Data projects

- Ongoing team mentoring
- Use case evaluation
- Management training
- Technical training
- Architecture reviews
- Live and email programming support

Go to <http://www.bigdatainstitute.io> for more information

About You

Your experience as a developer, analyst or administrator

Which languages you use

Experience with Hadoop, Big Data or NoSQL

Expectations from this class



Chapter 1

Introducing Apache Beam

- **What Is Beam?**
- Why Use Beam?
- Using Beam

Apache Beam

Apache Beam is a unified model for processing data

Was originally created at Google

- Later donated to the Apache Foundation as Apache Beam
- Now an Apache top level project

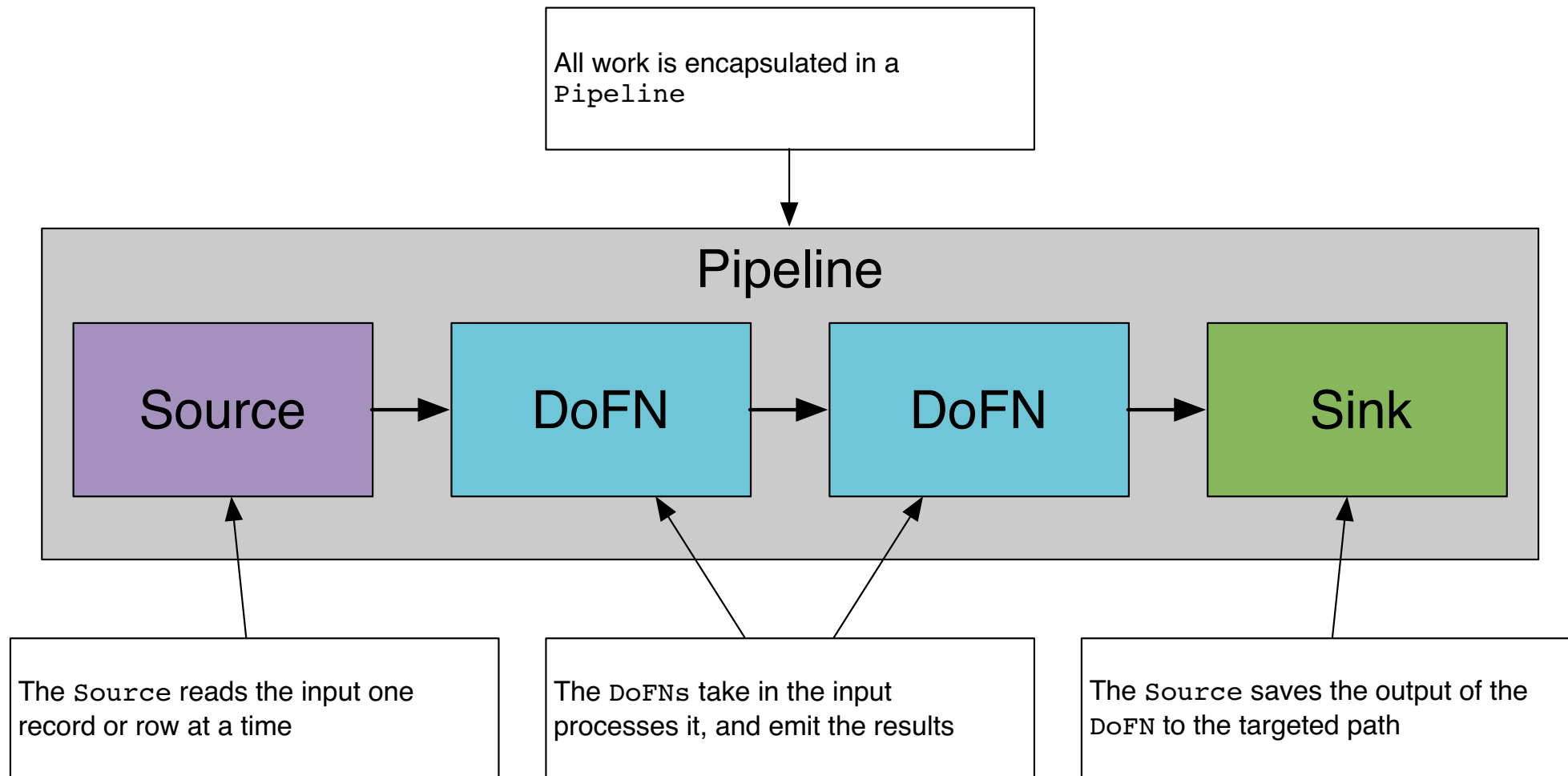
Beam code is written to its API

- Code is executed on different runners
- Not directly tied to a framework or runner

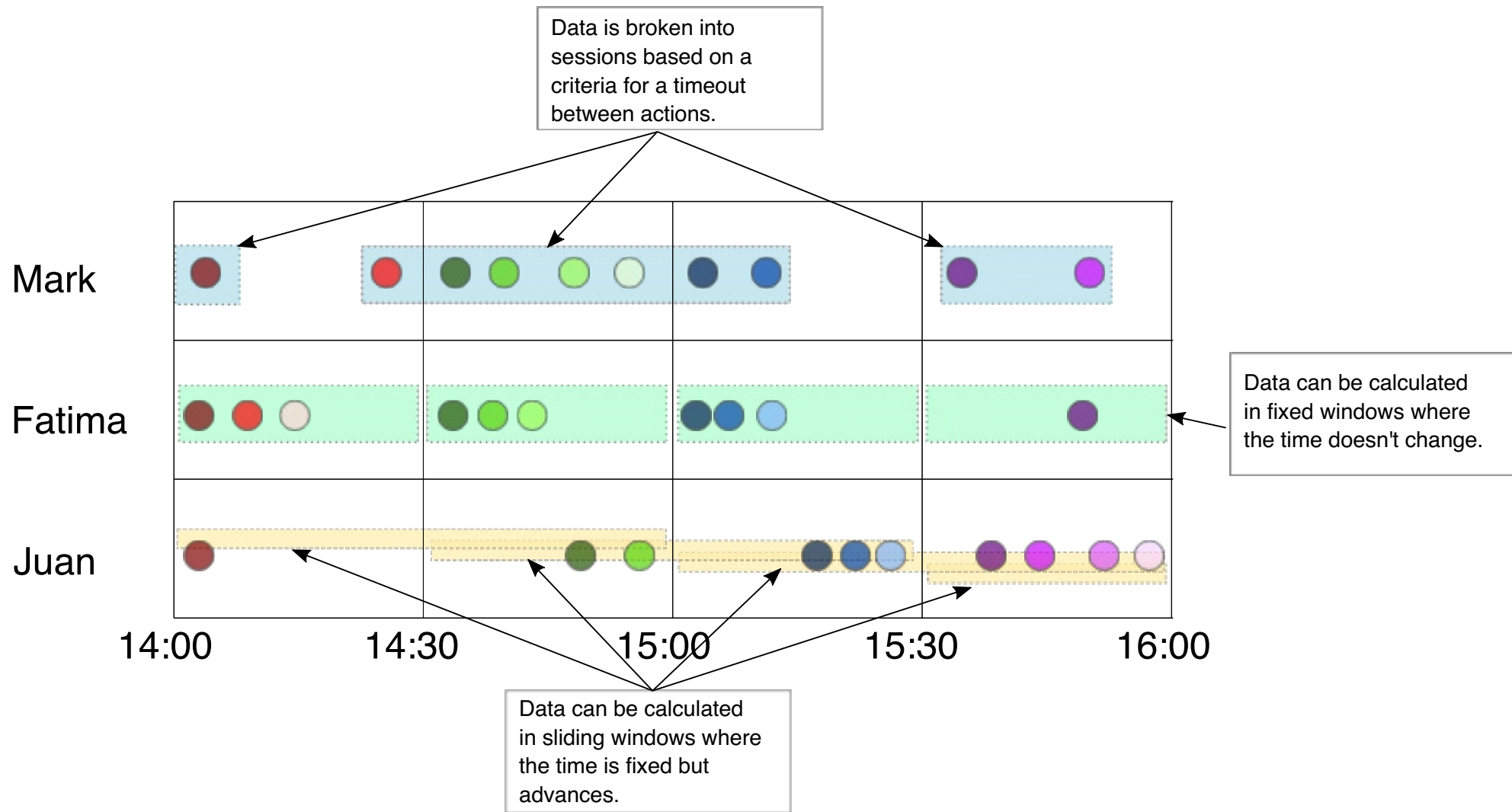
All interactions are done through pipelines



Beam Pipelines Diagram



Beam Windowing

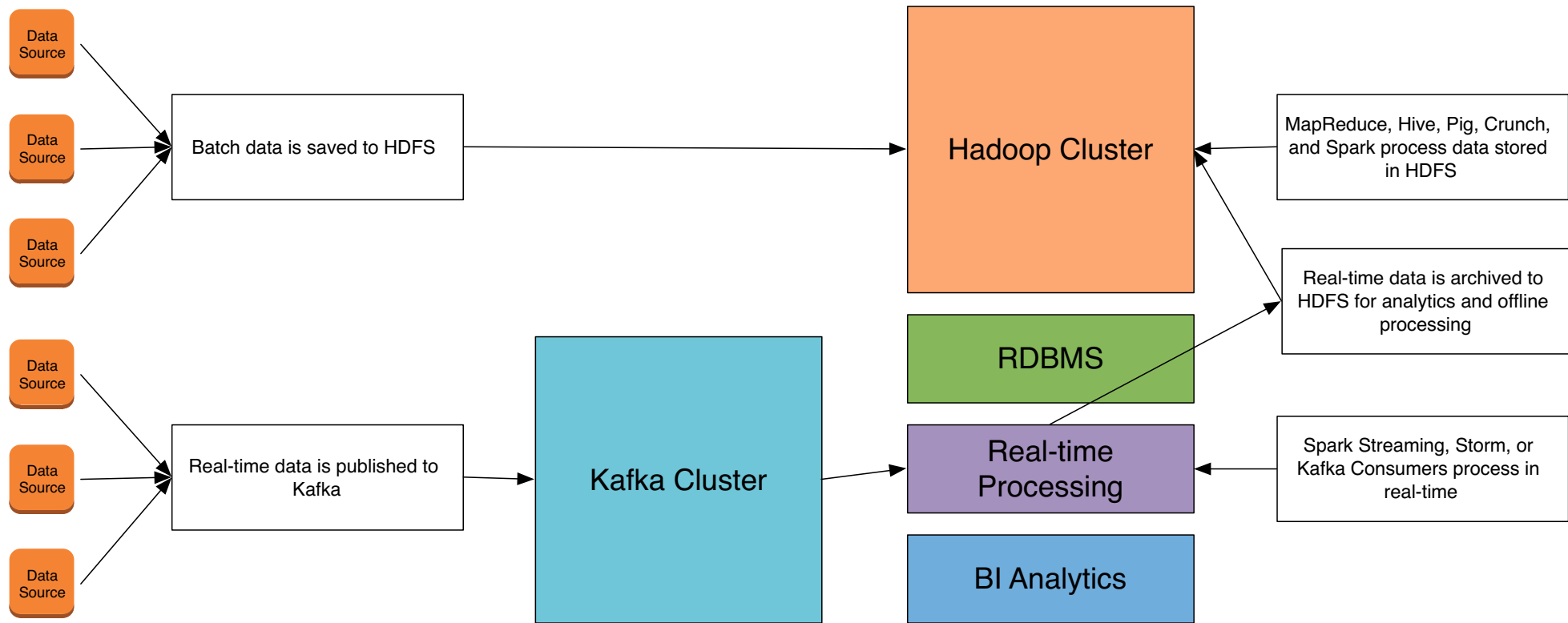


Introducing Apache Beam

- What Is Beam?
- **Why Use Beam?**
- Using Beam

Learning framework-specific APIs every time a new framework comes out or completely changes their existing API doesn't create value

General Architecture Diagram



Why I'm Excited About Beam

One API to rule them all

- One API to learn
- Move between frameworks

The most unified batch and stream API I've used

Unified API to the ecosystem

Risk mitigation of frameworks

Multiple languages

Running Beam

Beam isn't tied to a specific framework

Apache Spark uses the `spark-submit`

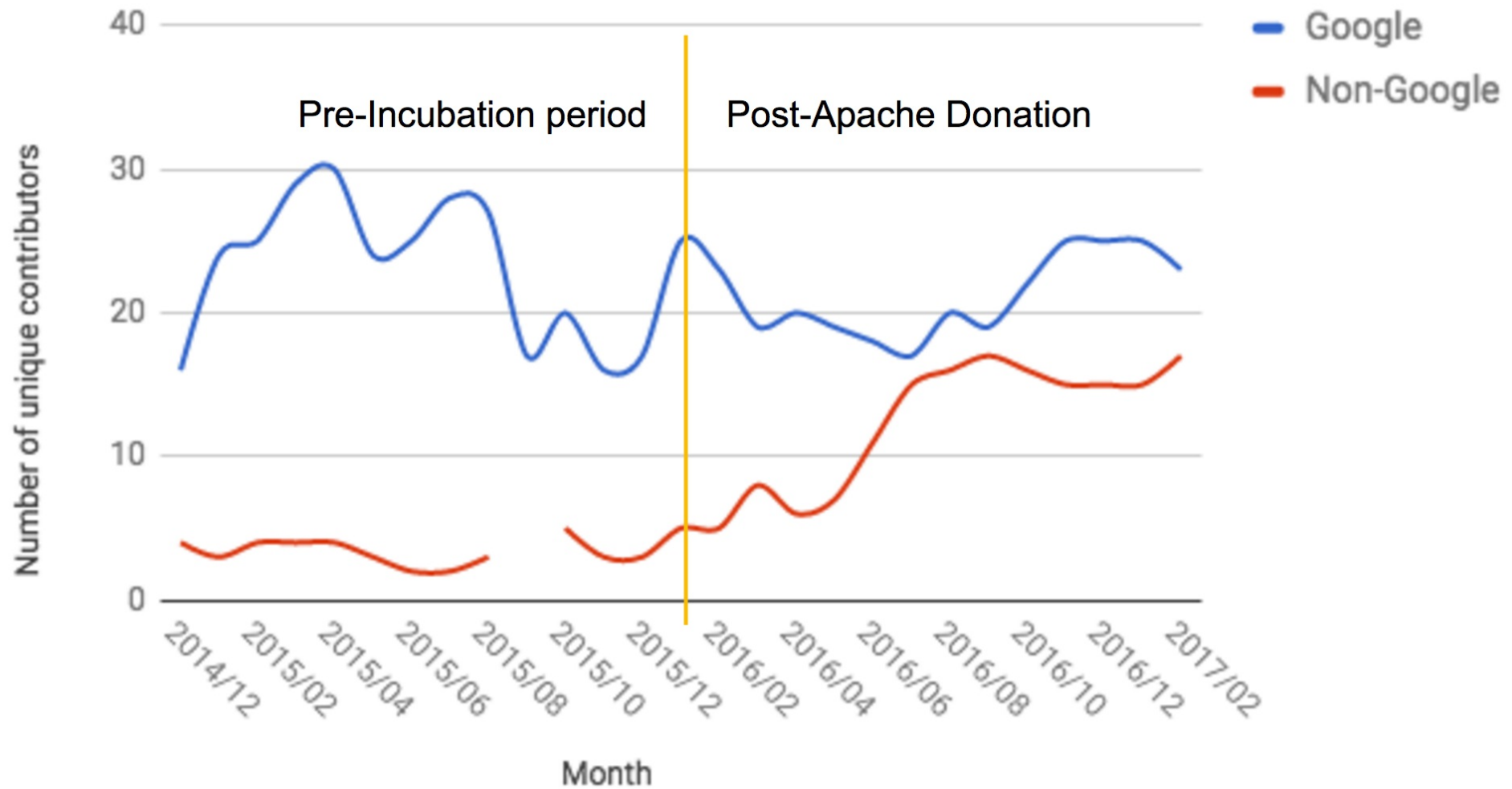
Apache Flink can be submitted with the Maven runner

Google Cloud Dataflow can be submitted with the Maven runner

The DirectRunner can be started with the Maven runner

Beam Contributions

Unique contributors per month



Introducing Apache Beam

- What Is Beam?
- Why Use Beam?
- **Using Beam**

MapElements

```
I  
cannot  
teach  
him  
The  
boy  
has  
no  
patience
```

```
PCollection<String> etl = lines.apply(MapElements.via(  
    (String line) -> line.toUpperCase()  
    ).withOutputType(TypeDescriptors.strings()));
```

```
I  
CANNOT  
TEACH  
HIM  
THE  
BOY  
HAS  
NO  
PATIENCE
```


Regex Transform

```
I cannot teach him. The boy has no patience.  
He will learn patience.
```

```
PCollection<String> linecount = lines.apply(Regex.matches("I.*\\."));
```

```
I cannot teach him. The boy has no patience.
```

Regular expressions can be used to parse KVs

```
I cannot teach him. The boy has no patience.  
He will learn patience.
```

```
PCollection<KV<String, String>> twoSentences =  
  lines.apply(Regex.findKV("(.*?)\\ (.*)", 1, 2));
```

```
<I cannot teach him, The boy has no patience>
```

Example Custom DoFN

```
I cannot teach him. The boy has no patience.  
He will learn patience.
```

```
PCollection<String> pats = lines.apply(ParDo.of(new PatLinesFN()));
```

```
static class PatLinesFN extends DoFn<String, String> {  
    @ProcessElement  
    public void processElement(DoFn<String, String>.ProcessContext  
                               context)  
        throws Exception {  
        String[] pieces = context.element().split(" ");  
  
        for (String piece : pieces) {  
            if (piece.startsWith("pat")) {  
                context.output(piece);  
            }  
        }  
    }  
}
```

```
patience.  
patience.
```

Playing Card Algorithm

```
import org.apache.beam.sdk.Pipeline;
import org.apache.beam.sdk.io.TextIO;
import org.apache.beam.sdk.options.PipelineOptions;
import org.apache.beam.sdk.options.PipelineOptionsFactory;
import org.apache.beam.sdk.transforms.Count;
import org.apache.beam.sdk.transforms.Regex;
import org.apache.beam.sdk.transforms.ToString;

public class PicoWordCount {
    public static void main(String[] args) {
        PipelineOptions options = PipelineOptionsFactory.create();
        Pipeline p = Pipeline.create(options);

        p
            .apply(TextIO.Read.from("playing_cards.tsv"))
            .apply(Regex.split("\\W+"))
            .apply(Count.perElement())
            .apply(ToString.elements())
            .apply(TextIO.Write.to("output/stringcounts"));

        p.run();
    }
}
```

Next Steps

What are other people doing with Beam?

- <http://tiny.jesse-anderson.com/beaminterview>

Where is some sample Beam code?

- <http://tiny.jesse-anderson.com/beamtutorial>

Main Beam site

- <https://beam.apache.org/>

Convincing your boss

- <http://tiny.jesse-anderson.com/beam1>
- <http://tiny.jesse-anderson.com/beam2>

About Me

Current: Instructor, Thought Leader, Monkey Tamer

Previously:

- Curriculum Developer and Instructor @ Cloudera
- Senior Software Engineer @ Intuit

Covered, Conferences and Published In:

- GigaOM, ArsTechnica, Pragmatic Programmers, Strata, OSCON, Wall Street Journal, CNN, BBC, NPR

See Me On:

- <http://www.jesse-anderson.com>
- [@jessetanderson](#)
- <http://tiny.bdi.io/linkedin>
- <http://tiny.bdi.io/youtube>