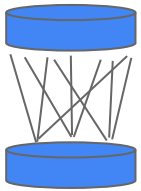
A Whirlwind Overview of Apache Beam

Eugene Kirpichov <kirpichov@google.com>
Staff Software Engineer

Google Cloud Platform

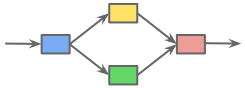
(2004) MapReduce

SELECT + GROUPBY



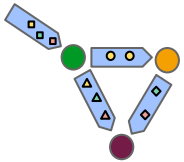
(2008) FlumeJava

High-level API



(2013) Millwheel

Deterministic
Streaming

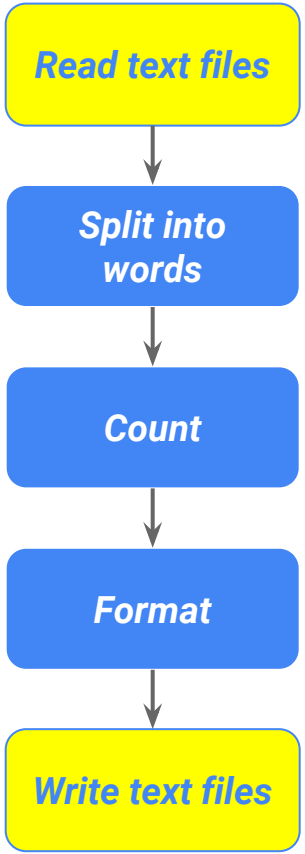


(2014) Dataflow

Unified batch/streaming,
Portable

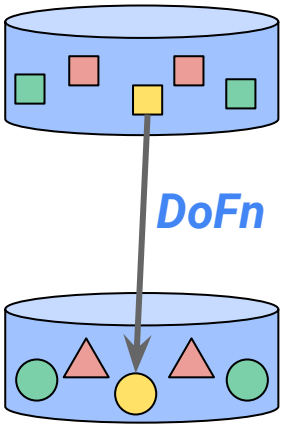
(2016)
Apache Beam

Open ecosystem,
Community-driven
Vendor-independent

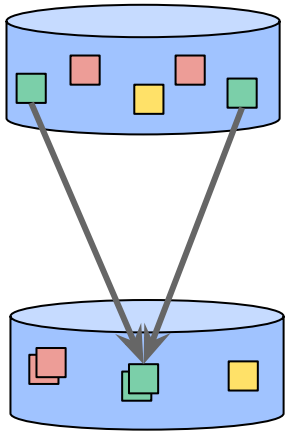



```
Pipeline p = Pipeline.create(options);

PCollection<String> lines = p.apply(
    TextIO.read().from("gs://.../*"));
PCollection<KV<String, Long>> wordCounts = lines
  .apply(FlatMapElements.via(word → word.split("\\W+")))
  .apply(Count.perElement());
wordCounts
  .apply(MapElements.via(
     count → count.getKey() + ": " + count.getValue()))
  .apply(TextIO.write().to("gs://.../..."));

p.run();
```

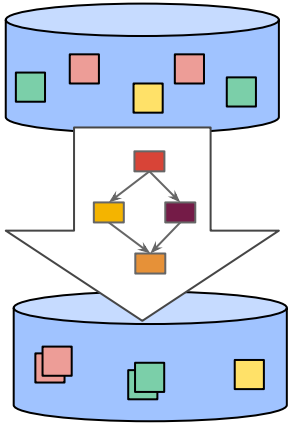
Beam PTransforms



ParDo
("map")

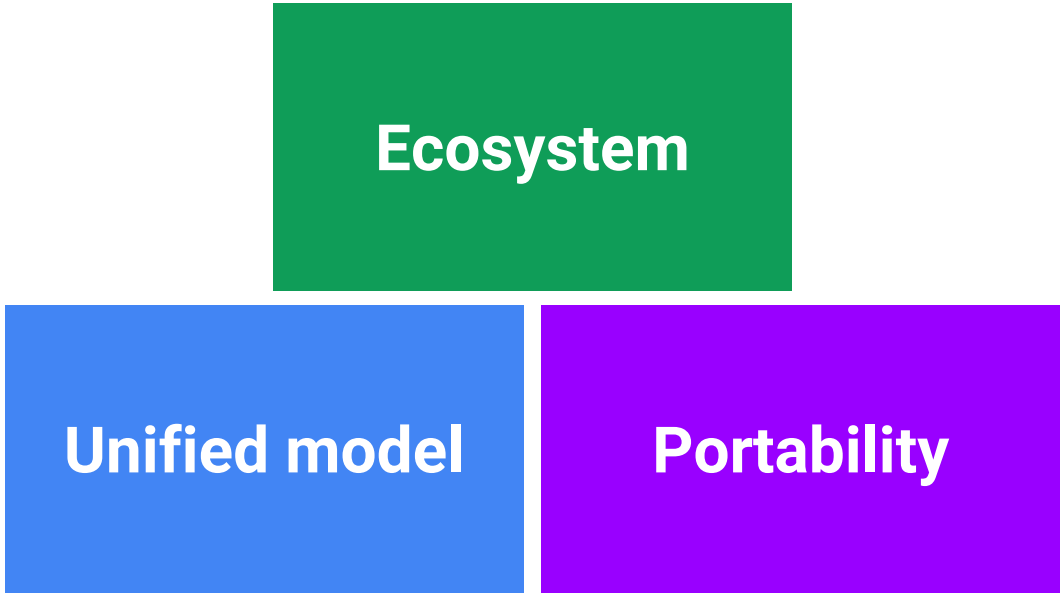


GroupByKey
("reduce")



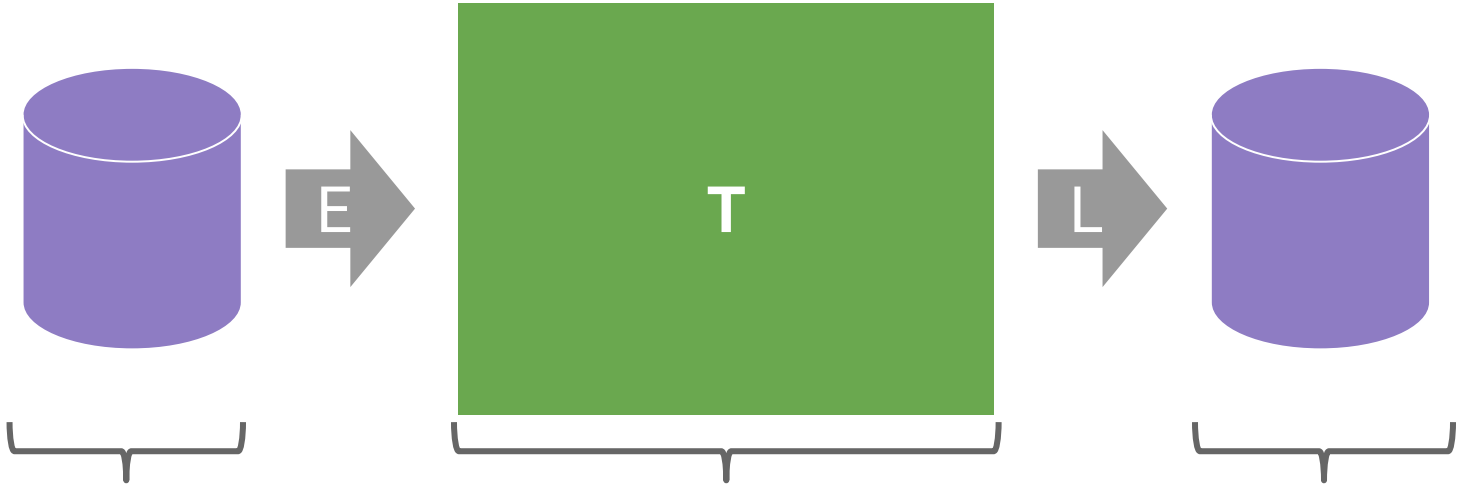
Composite

Pillars of Beam



Unified Model

Batch doesn't exist



Grows

(Always expect new data)

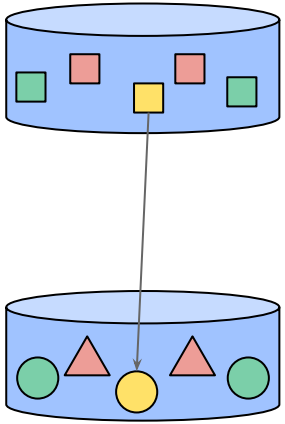
Computes updates

Evolves

Growing data is temporal ⇒ All data has timestamps (**event-time:** $t_{happened}$)

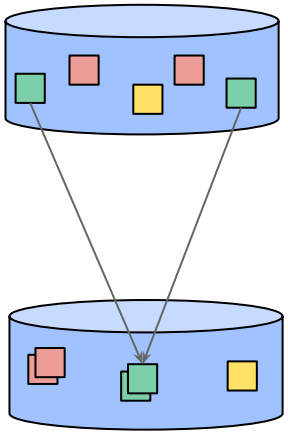
Dealing with new data

ParDo



⇒ **Apply to new data**

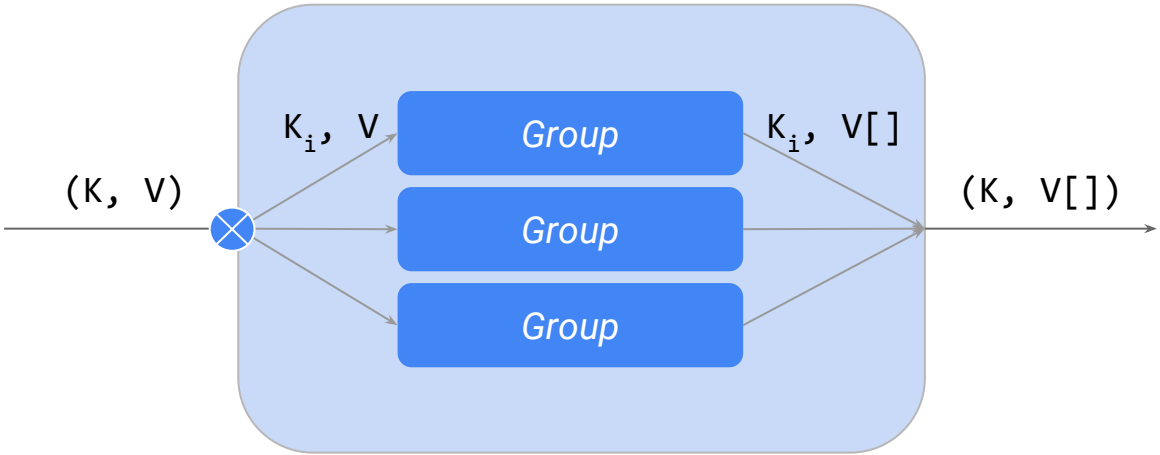
GroupByKey

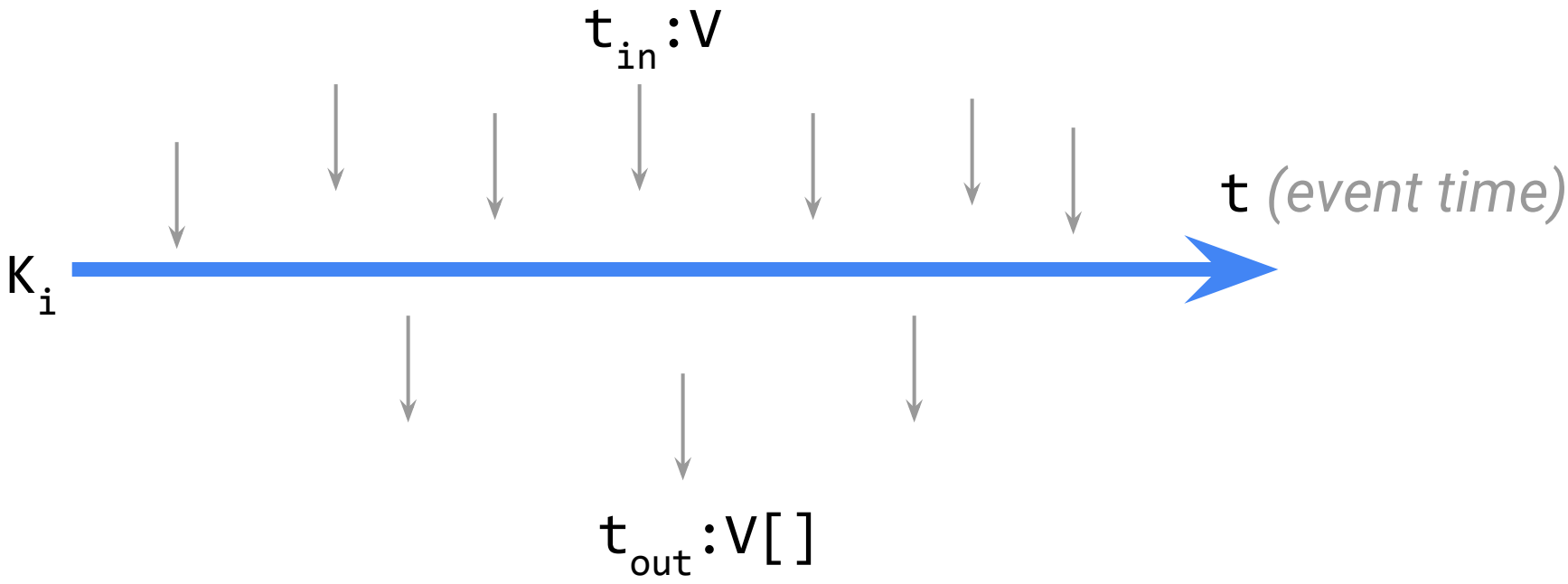


⇒ **?**

Continuous aggregation

Idea: per-key buffering

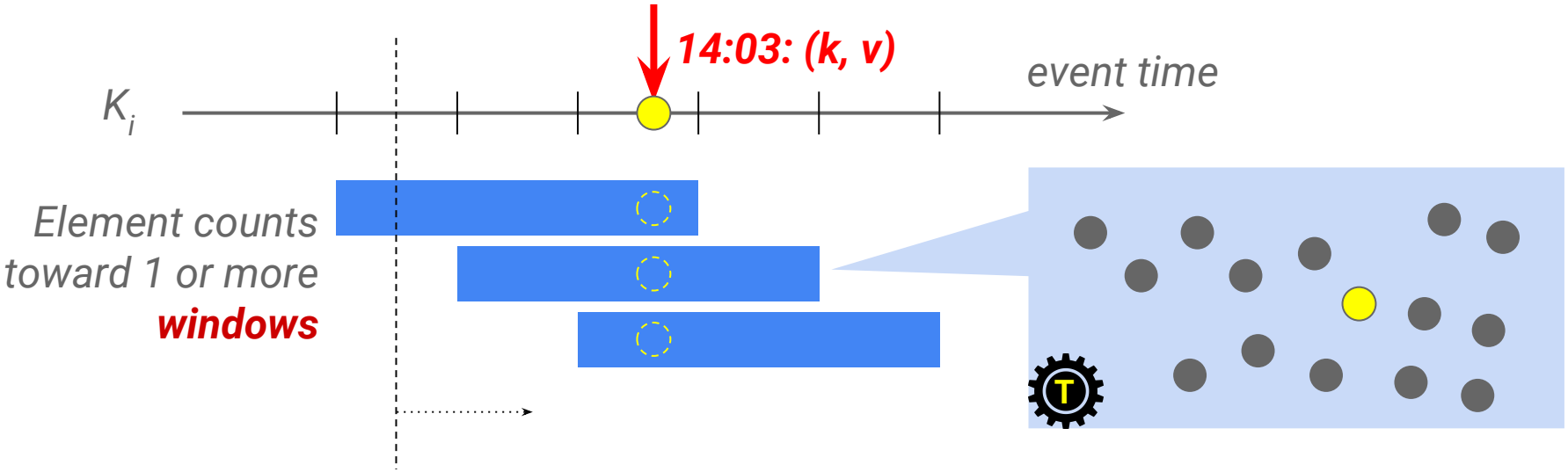




See: Streams and Tables
https://www.infoq.com/presentations/beam-model-stream-table=theory

Continuous aggregation

Idea: temporal windowing



watermark
closes old windows

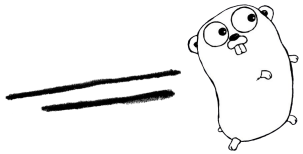
Apply (user-specified) ***trigger*** ⇒
drop / add to buffer / emit buffer

There is no batch / streaming.

Only different ways to control aggregation

Portability

(vision for 2018)



• • •

Code in any supported language *(or a mix)*

Portable pipeline representation

• • •

Run on any supported runner

No vendor lock-in

Run any language on any runner

No language lock-in

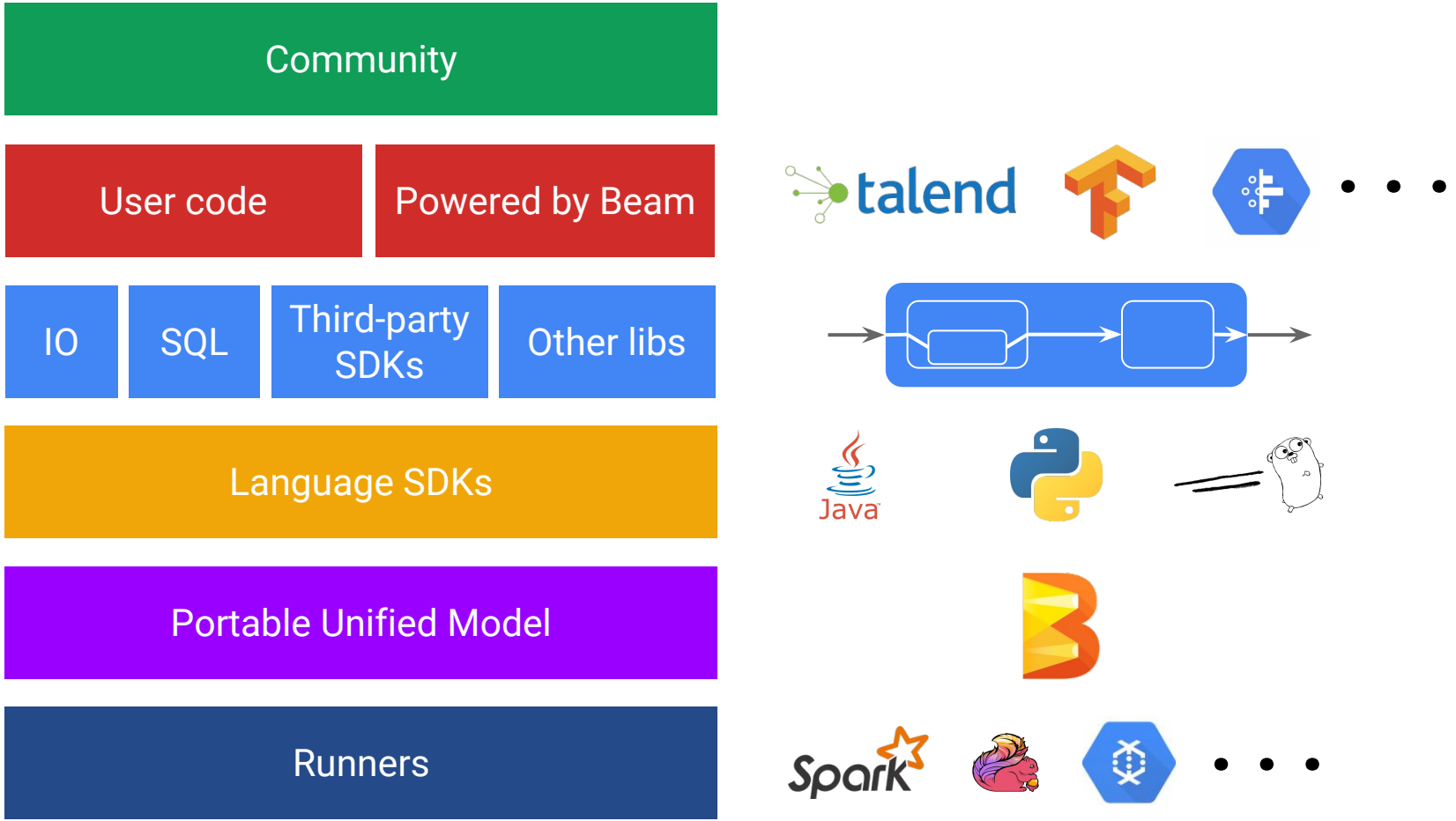
Users: Use all transforms from all languages

Library authors: Will be usable by all languages

Accelerated ecosystem growth

New runner / new SDK ⇒ access all Beam libraries

Ecosystem


Community
User code
Powered by Beam
IO
SQL
Third-party SDKs
Other libs
Language SDKs
Portable Unified Model
Runners
talend
Java
Spark

250 contributors

31 committers (**11** orgs)

~5000 PRs

~12,500 commits

25+ IO connectors

5 stable releases

9 runners

Thank you!