Beam Concepts Review

Israel Herraiz

Strategic Cloud Engineer,
Google Cloud

Agenda

Course Intro

Beam Concepts Review

Windows, Watermarks, and Triggers

Sources and Sinks

Schemas

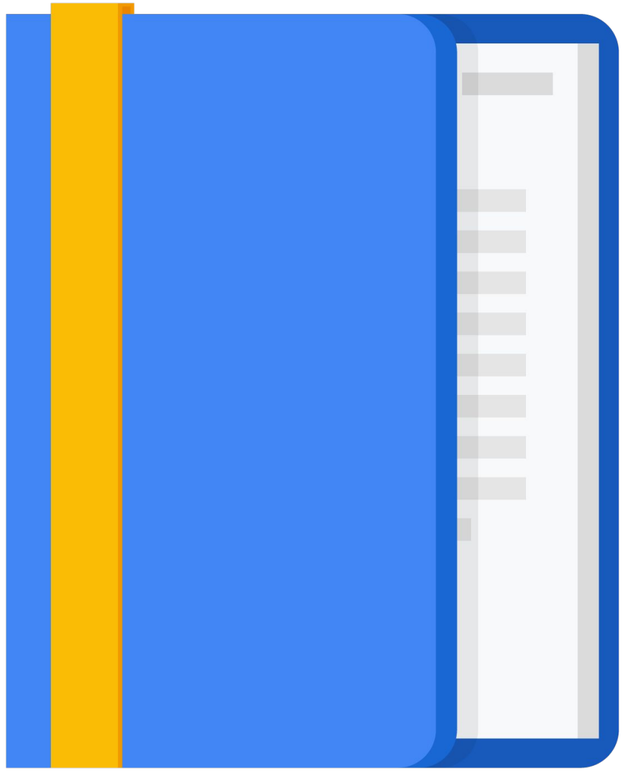
State and Timer

Best Practices

SQL and DataFrames

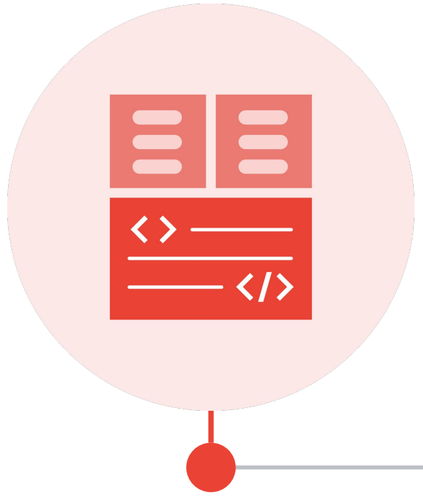
Beam Notebooks

Summary



Beam concepts review

Agenda

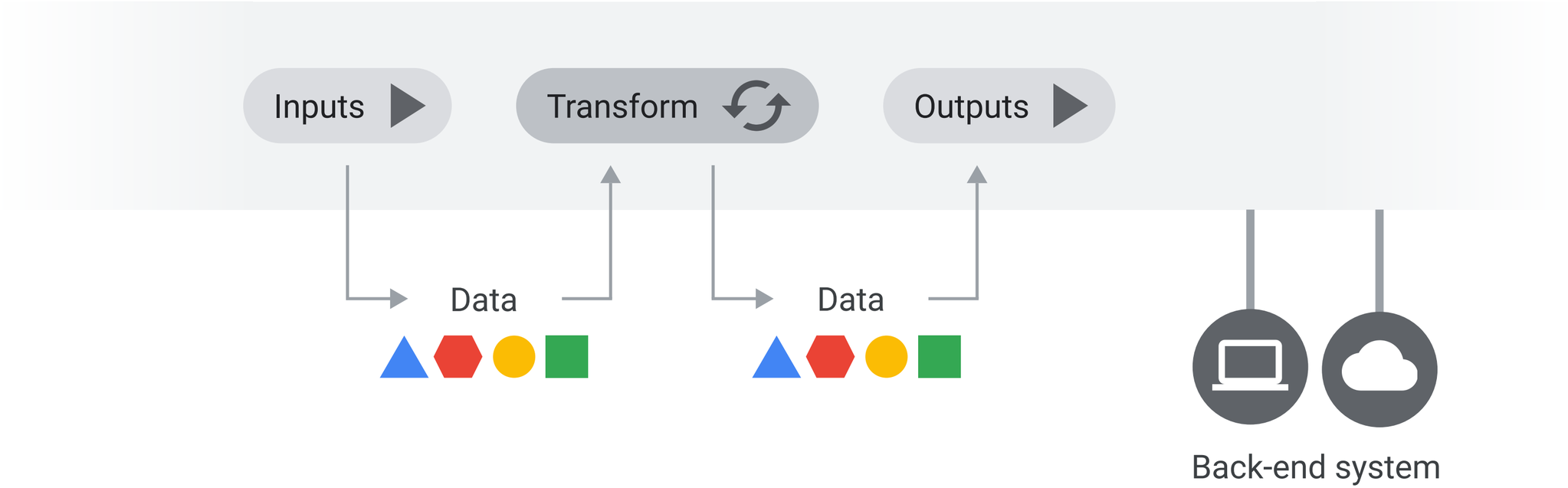


Beam basics

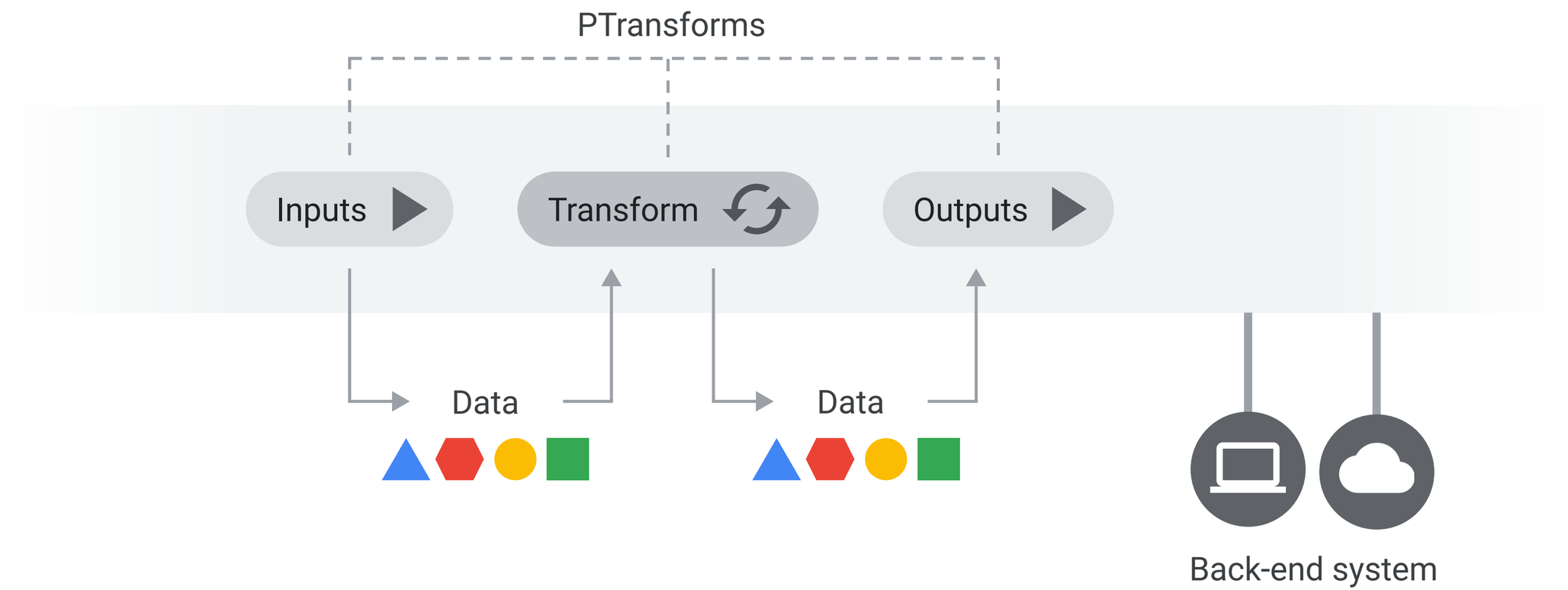
Utility transforms

DoFn lifecycle

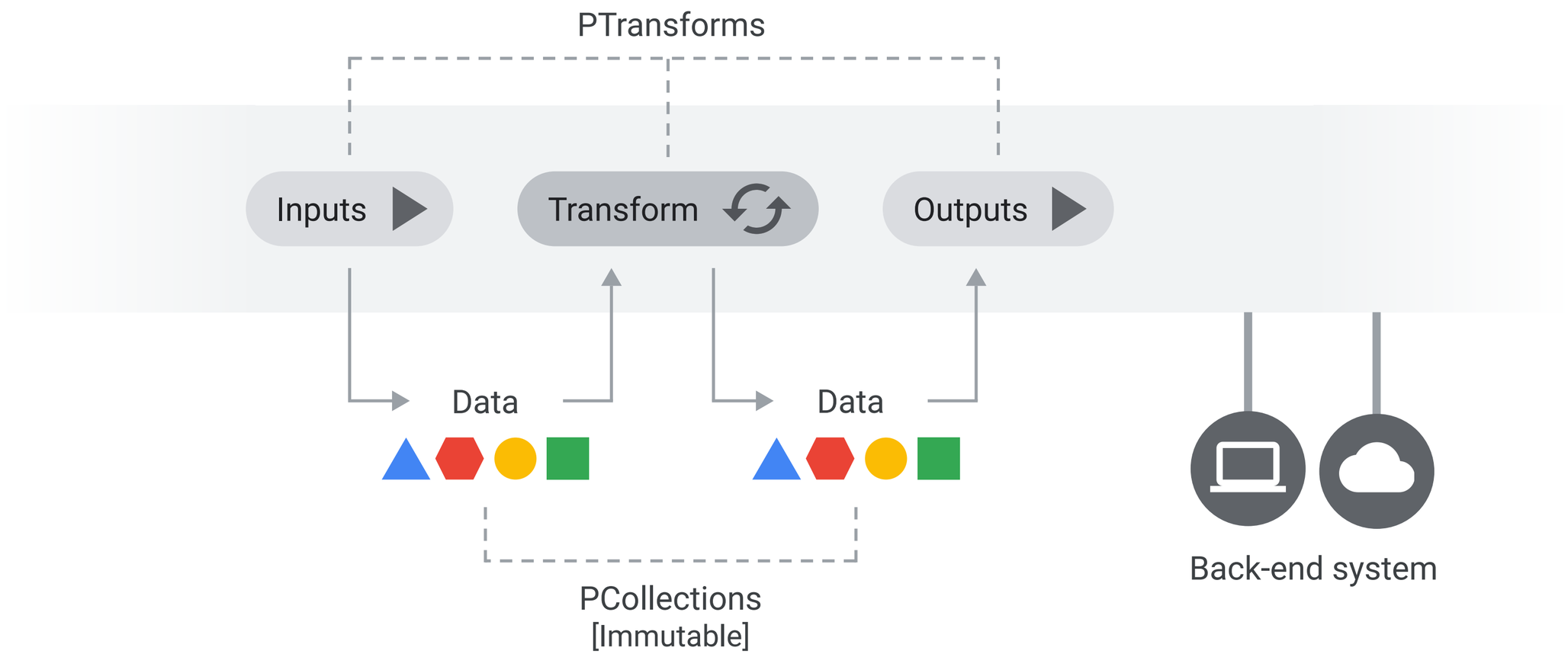
Apache Beam = Batch + stream



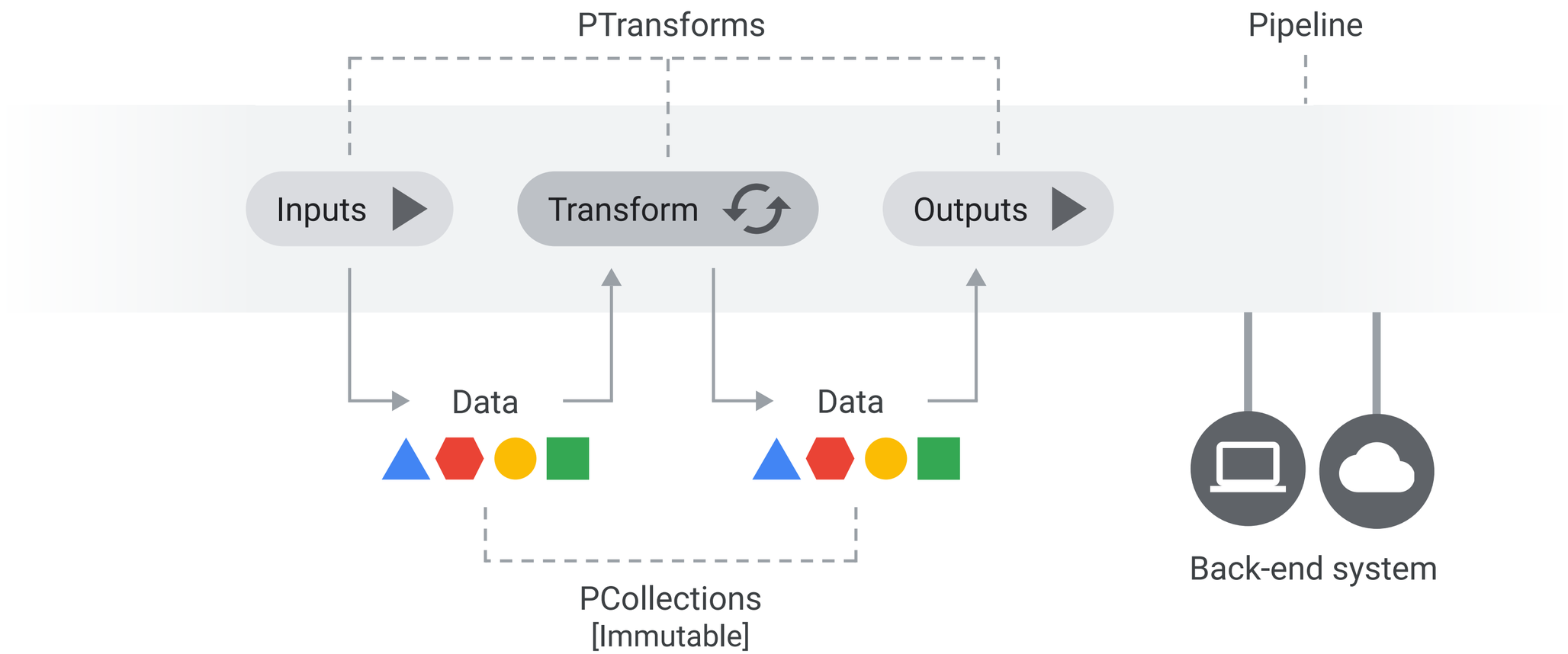
Apache Beam = Batch + stream



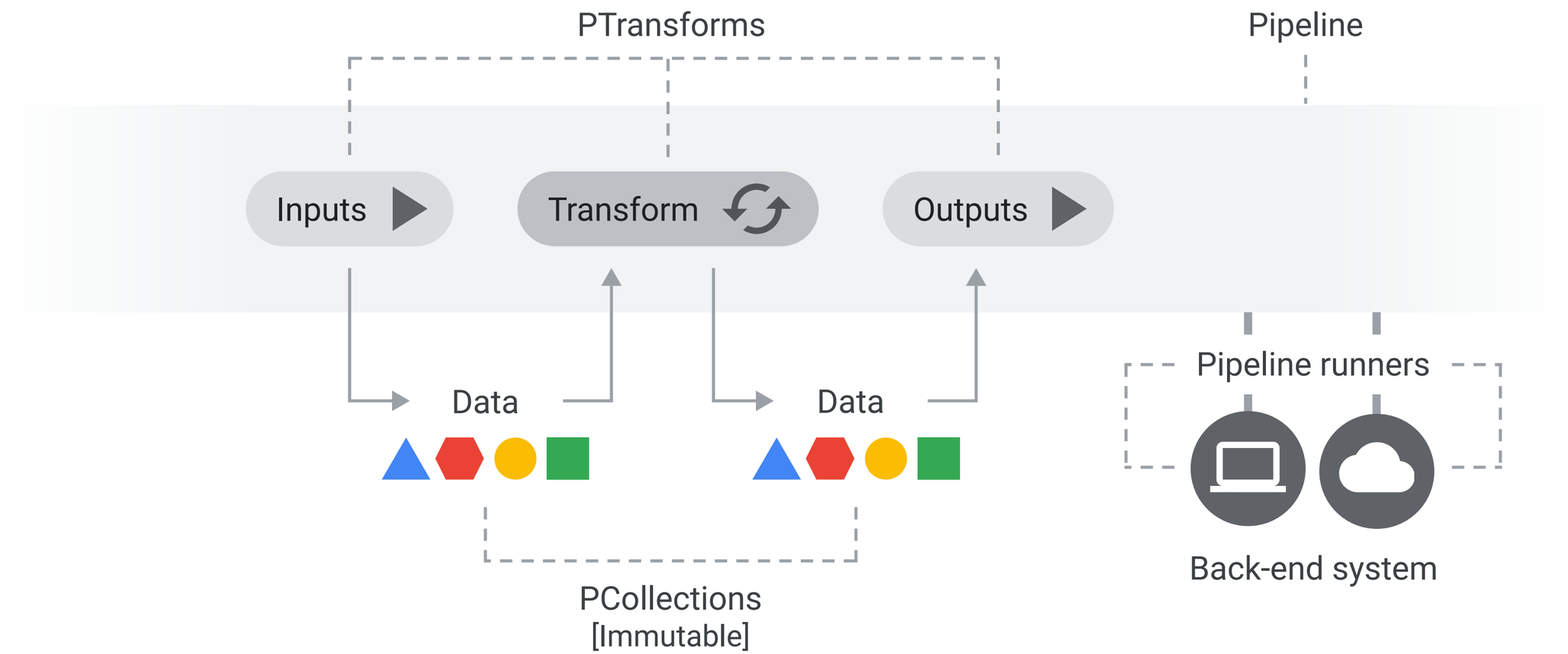
Apache Beam = Batch + stream



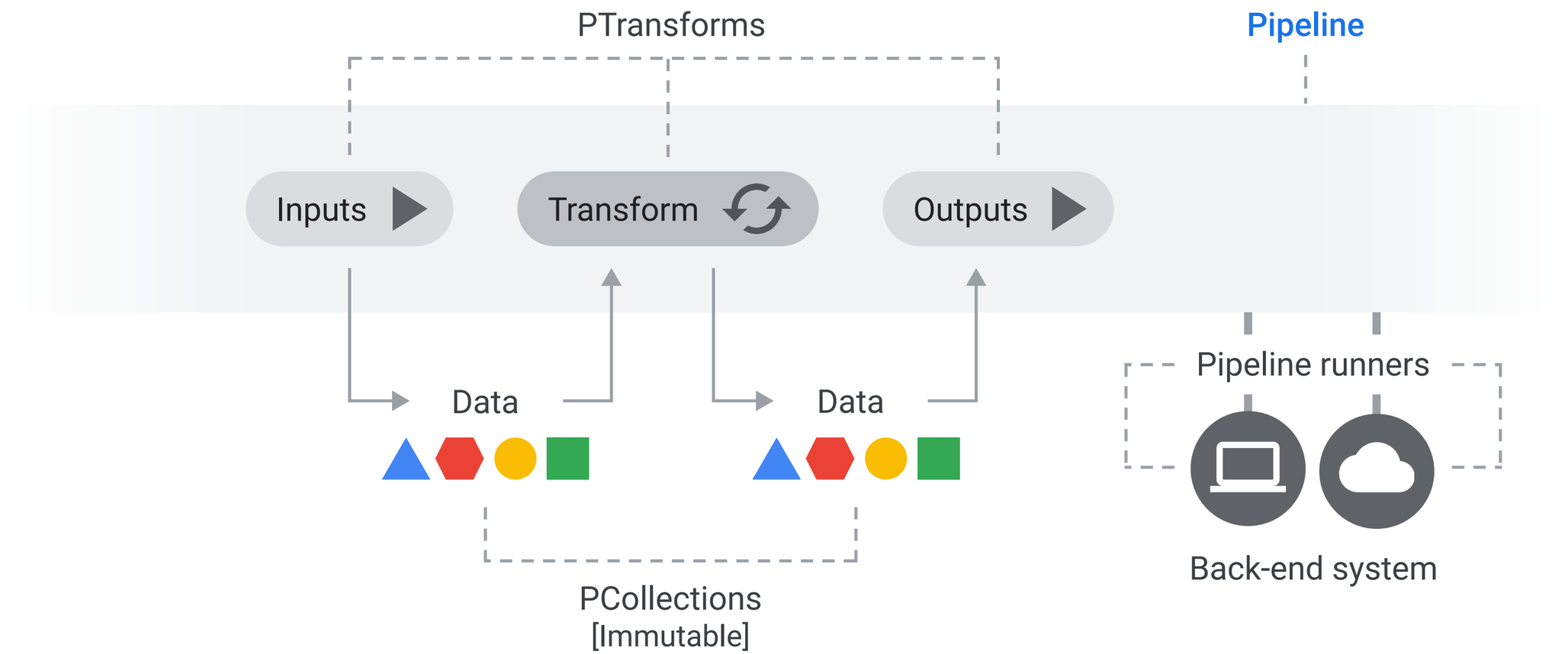
Apache Beam = Batch + stream



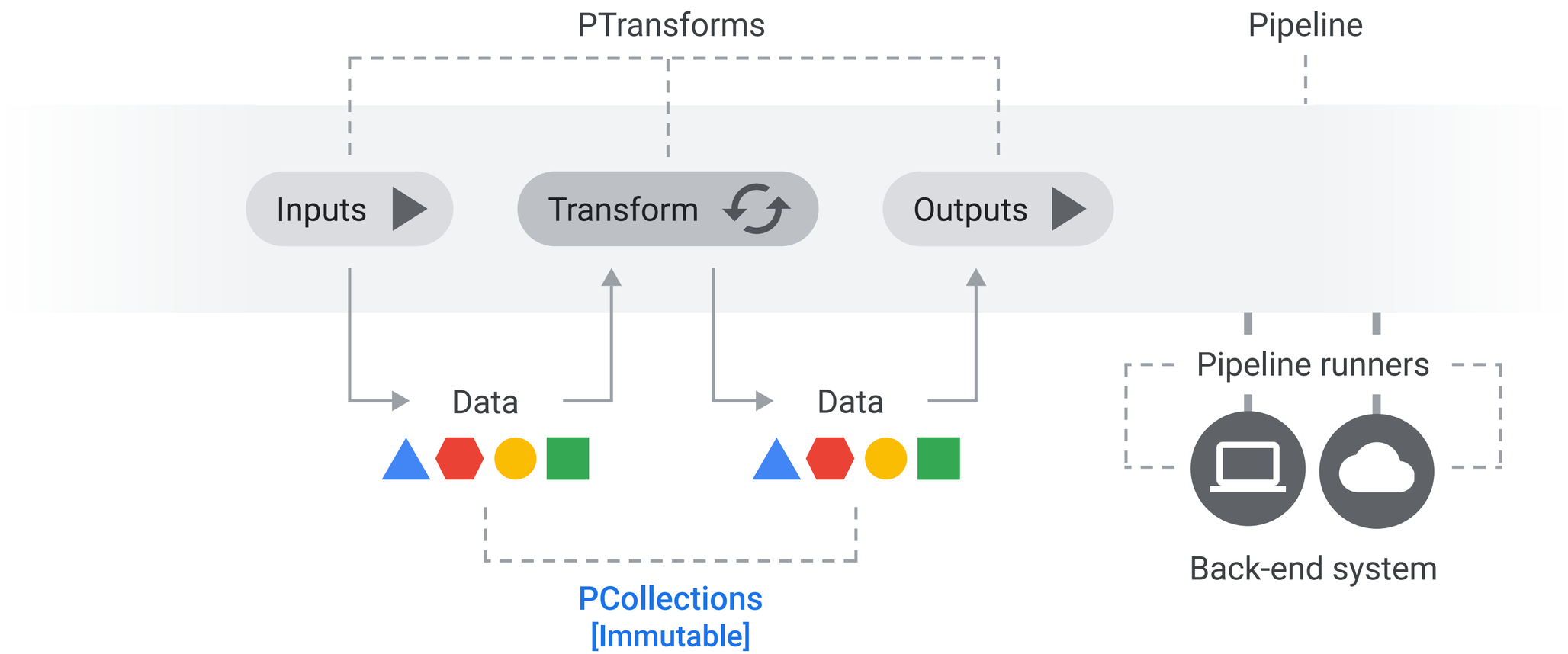
Apache Beam = Batch + stream



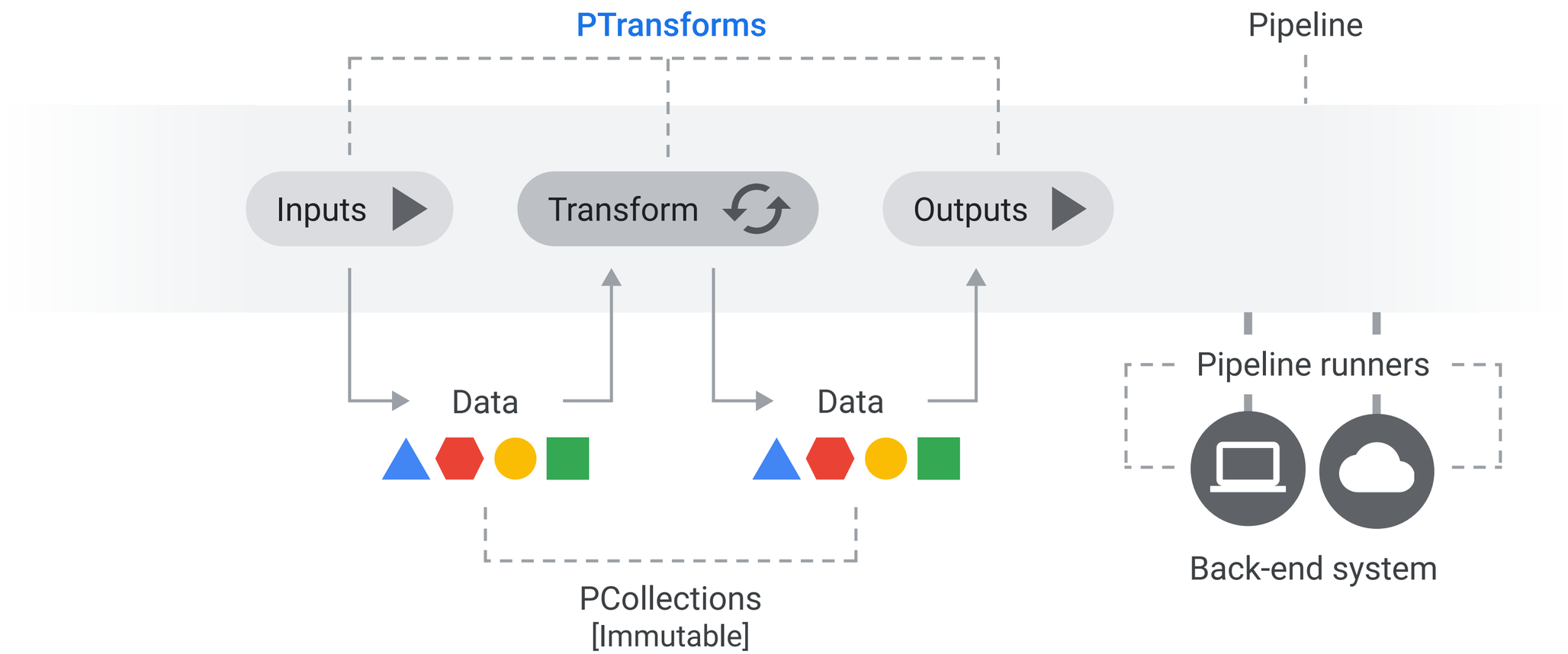
Apache Beam = Batch + stream



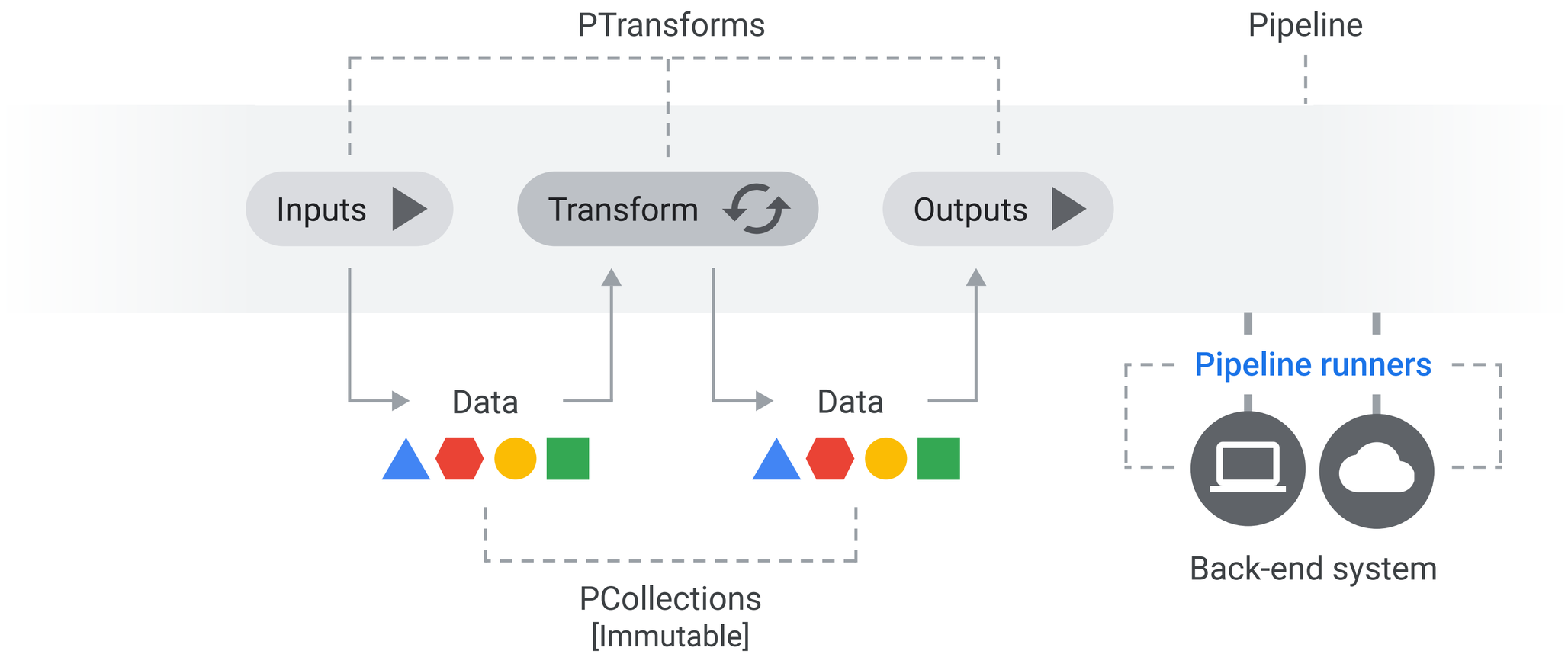
Apache Beam = Batch + stream



Apache Beam = Batch + stream

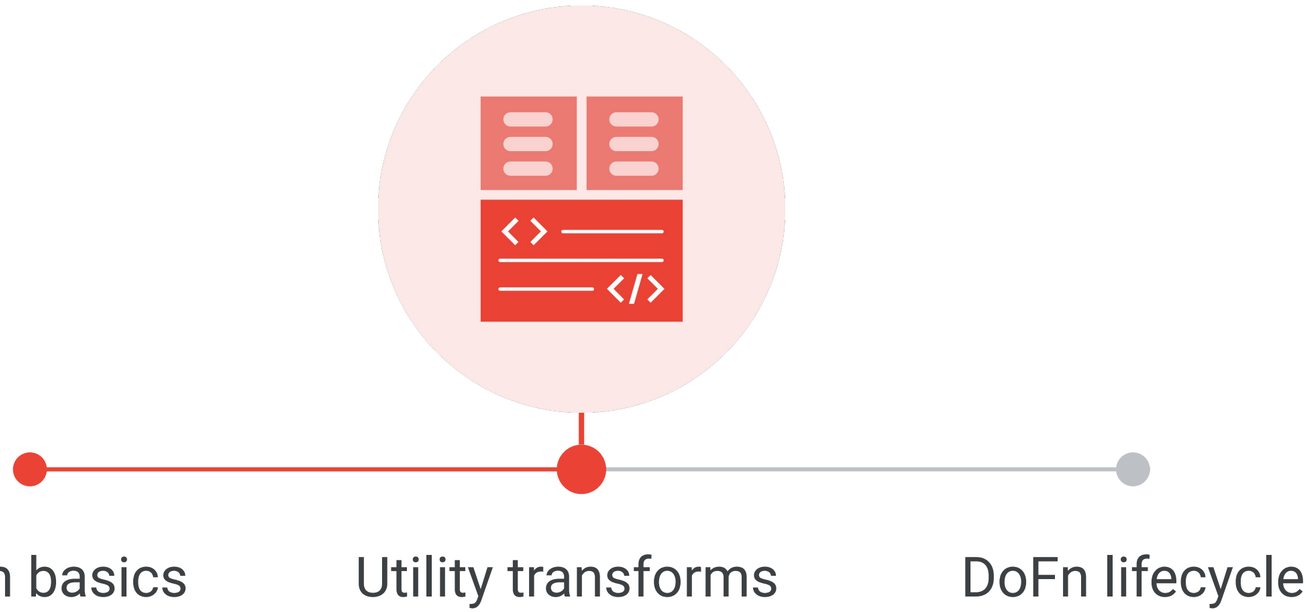


Apache Beam = Batch + stream



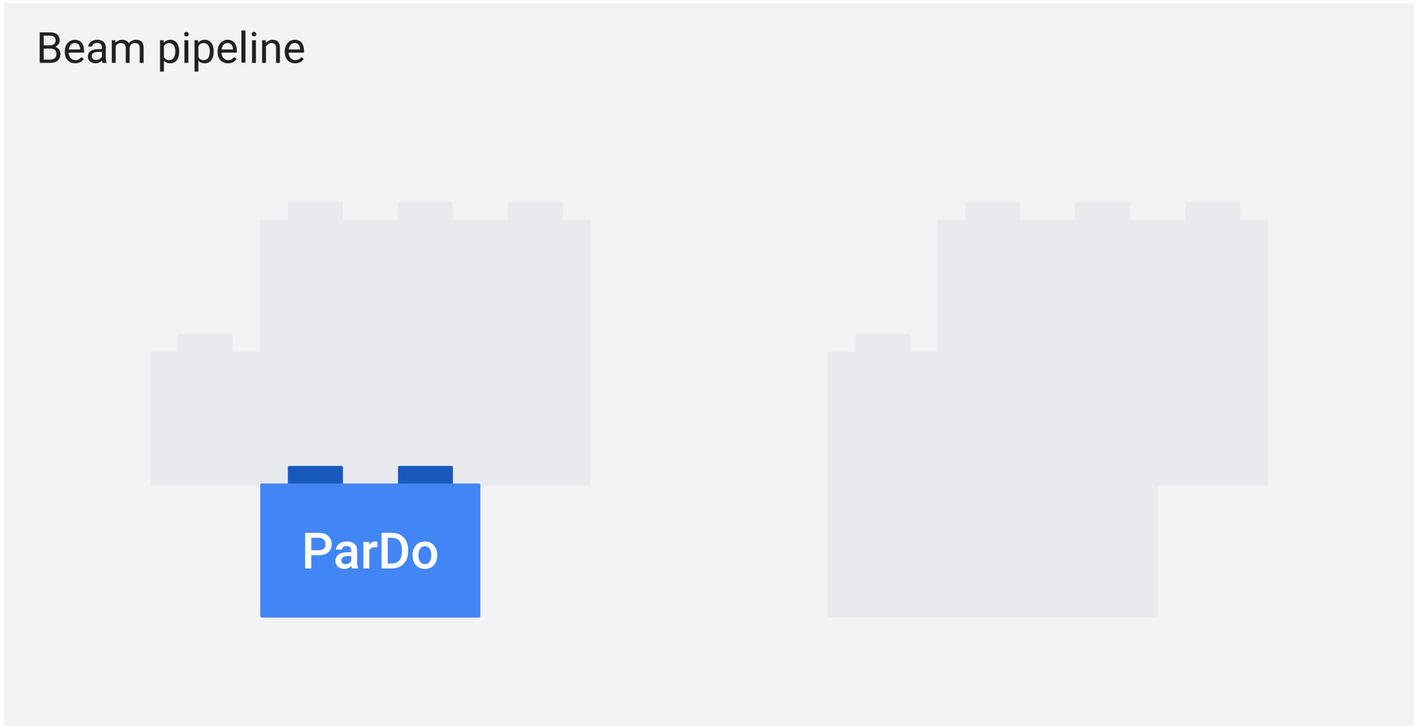
Beam concepts review

Agenda

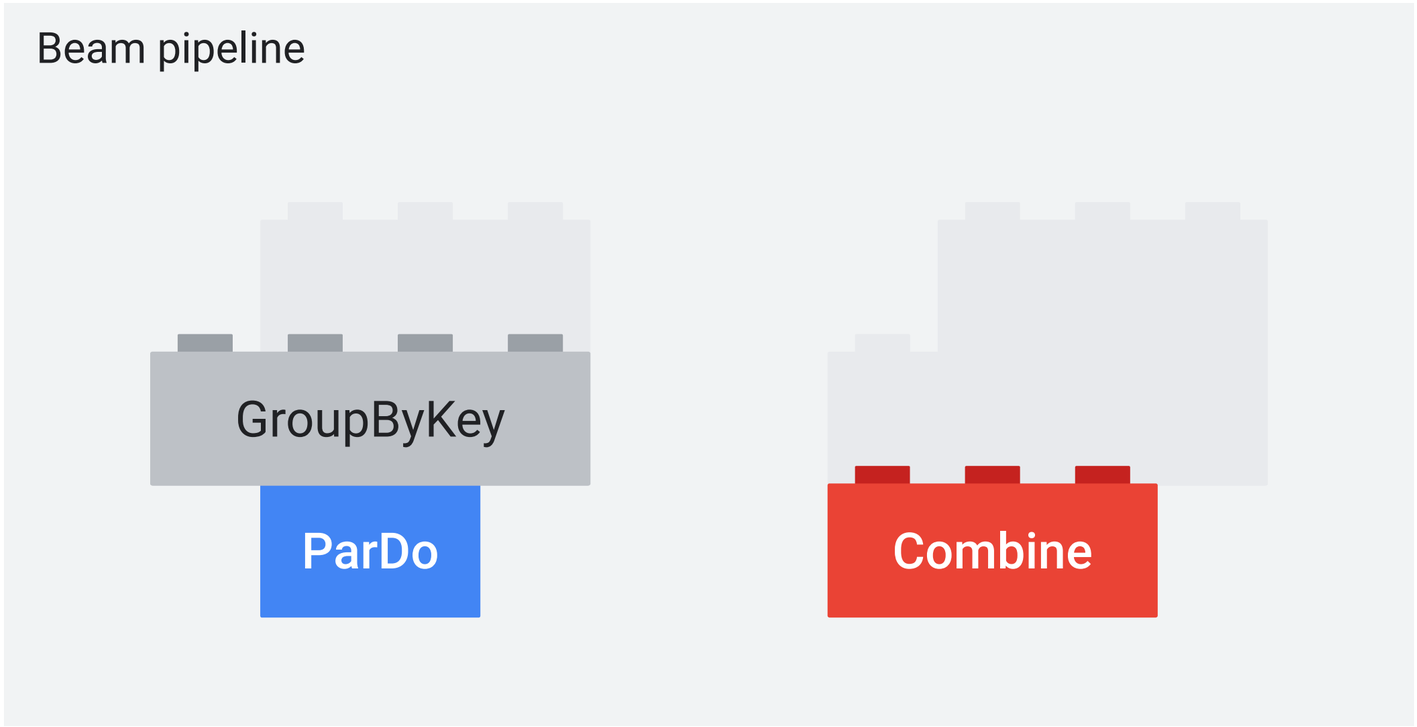


Transforms

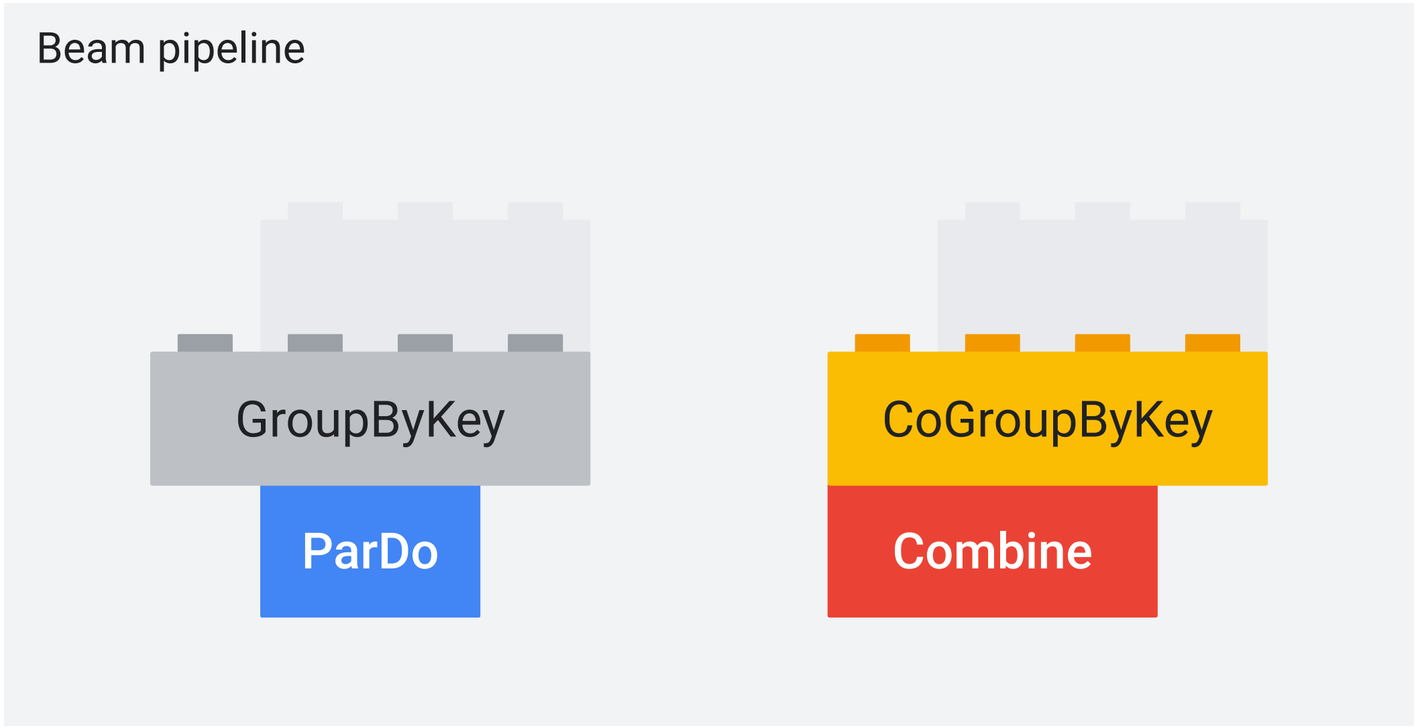
Transforms



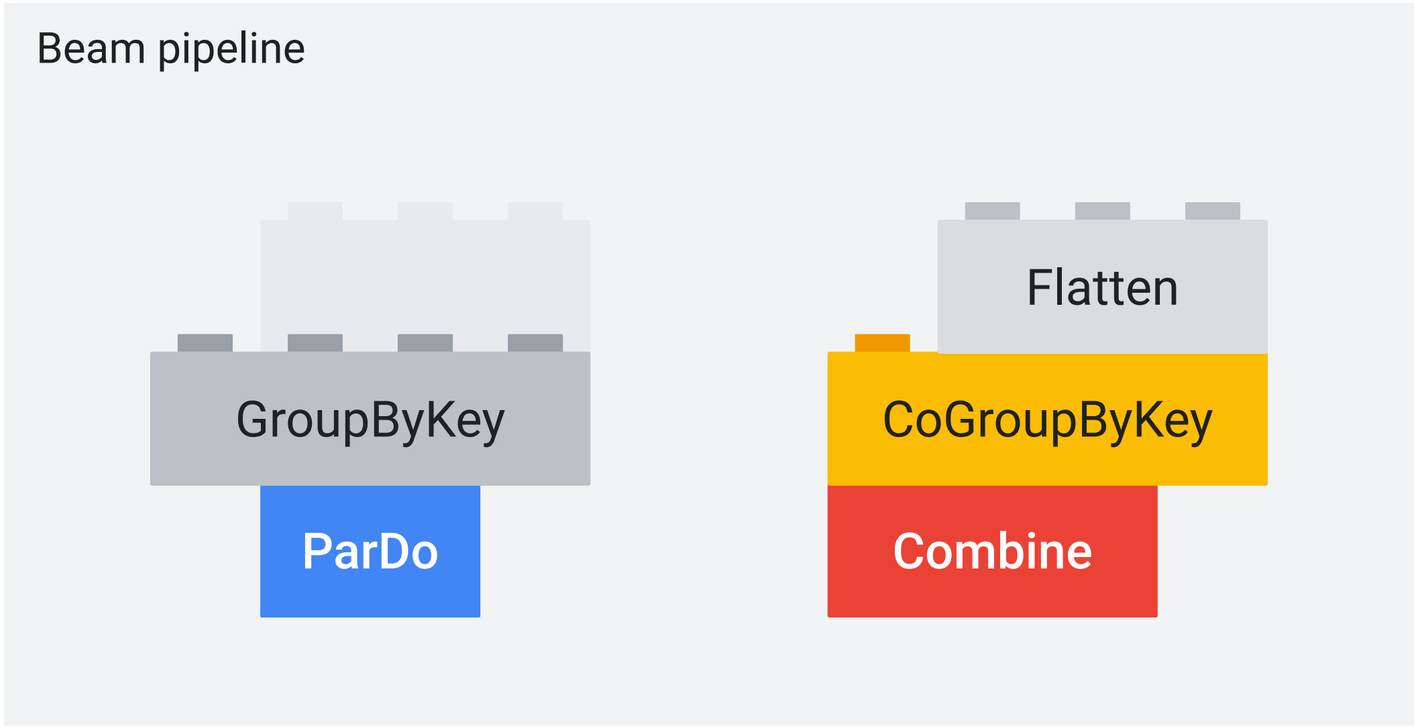
Transforms



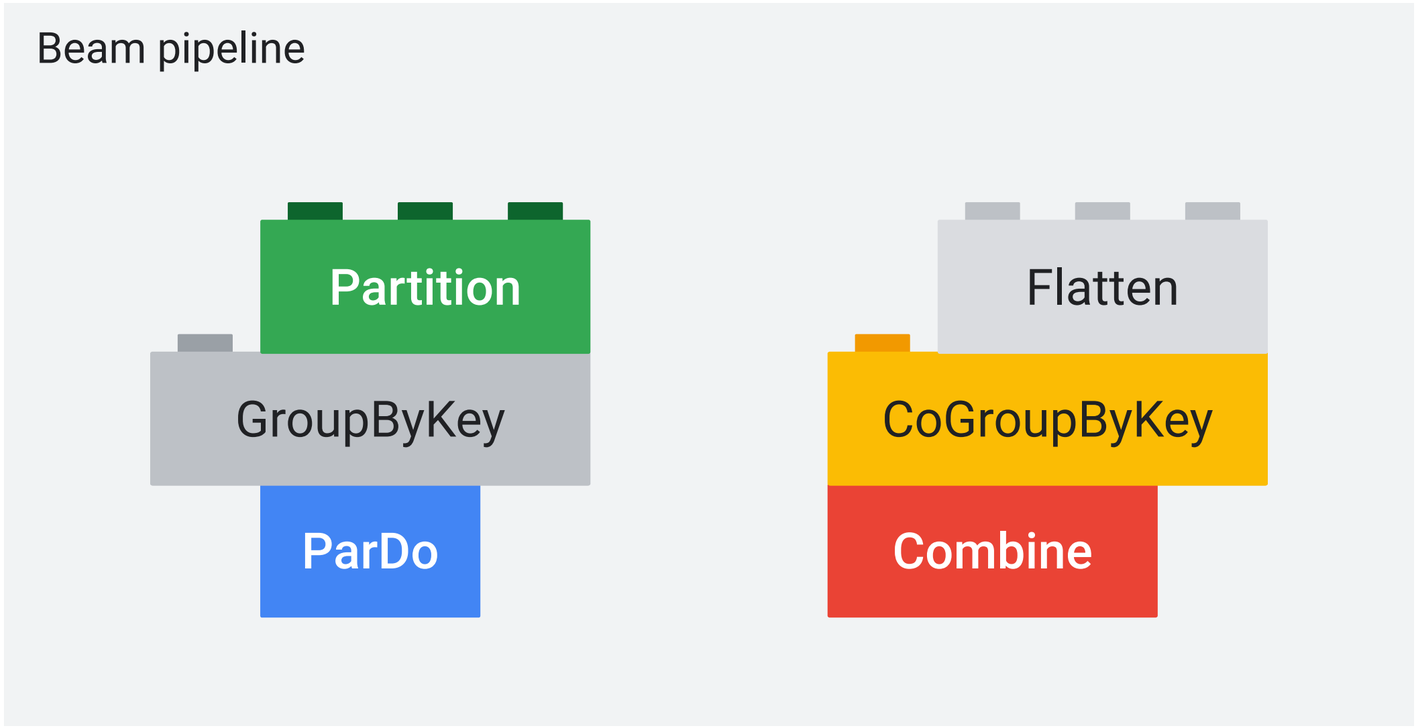
Transforms



Transforms

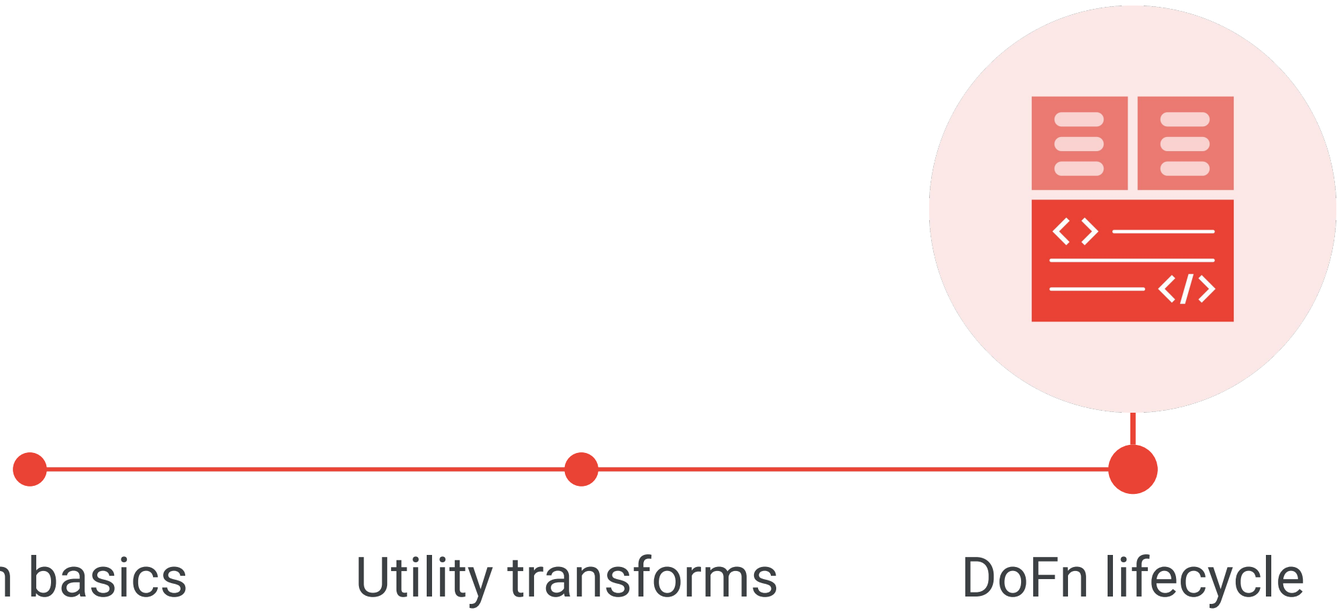

Beam pipeline
GroupByKey
ParDo
Flatten
CoGroupByKey
Combine

Transforms


Beam pipeline
Partition
GroupByKey
ParDo
Flatten
CoGroupByKey
Combine

Beam concepts review

Agenda



Friends of ParDo

	Input	Output	Side inputs and side outputs
ParDo	**1**	**0, 1 or many**	✓
Filter	1	0 or 1	✗
MapElements	1	1	✗
FlatMapElements	1	0, 1 or Many	✗
WithKeys	value	(f(value), value)	✗
Keys	(key, value)	key	✗
Values	(key, value)	value	✗

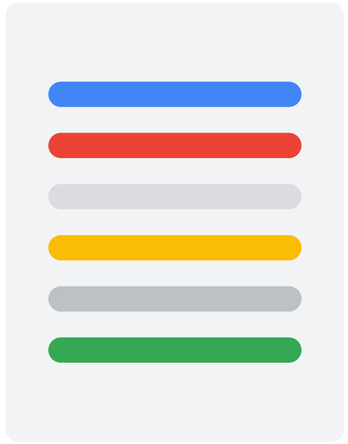
Friends of ParDo

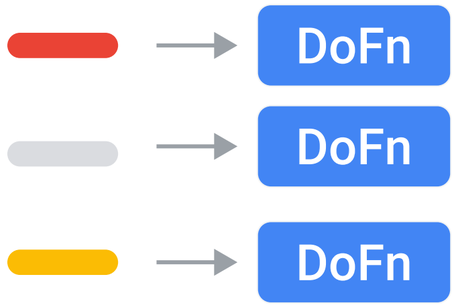
	Input	Output	Side inputs and side outputs
ParDo	**1**	**0, 1 or many**	✓
Filter	1	0 or 1	✗
MapElements	**1**	**1**	✗
FlatMapElements	1	0, 1 or Many	✗
WithKeys	value	(f(value), value)	✗
Keys	(key, value)	key	✗
Values	(key, value)	value	✗

Friends of ParDo

	Input	Output	Side inputs and side outputs
ParDo	**1**	**0, 1 or many**	✓
Filter	1	0 or 1	✗
MapElements	**1**	**1**	✗
FlatMapElements	**1**	**0, 1 or Many**	✗
WithKeys	value	(f(value), value)	✗
Keys	(key, value)	key	✗
Values	(key, value)	value	✗

Data bundles





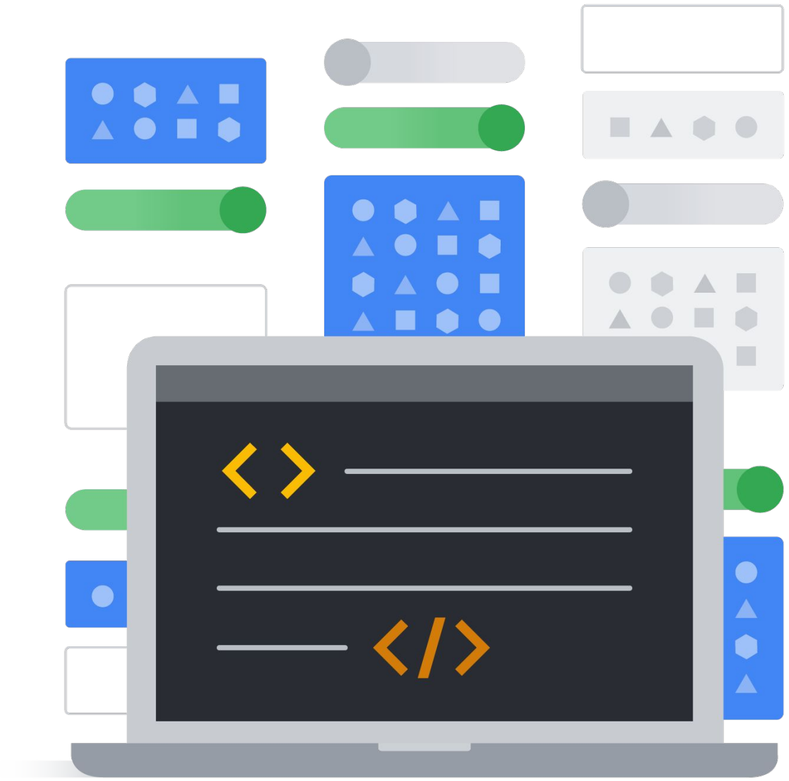
Worker receives bundles of work

Each bundle is passed to a DoFn object

Many DoFns can be running at the same time within the same process

Methods of DoFn

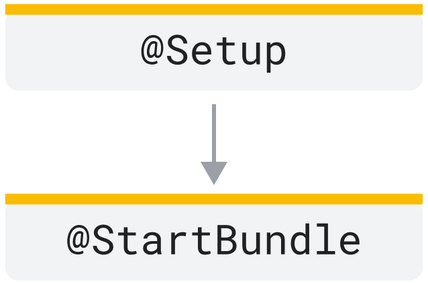
```
class MyDoFn(beam.DoFn):
    def setup(self):
        pass
    def start_bundle(self):
        pass
    def process(self, element):
        pass
    def finish_bundle(self):
        pass
    def teardown(self):
        pass
```



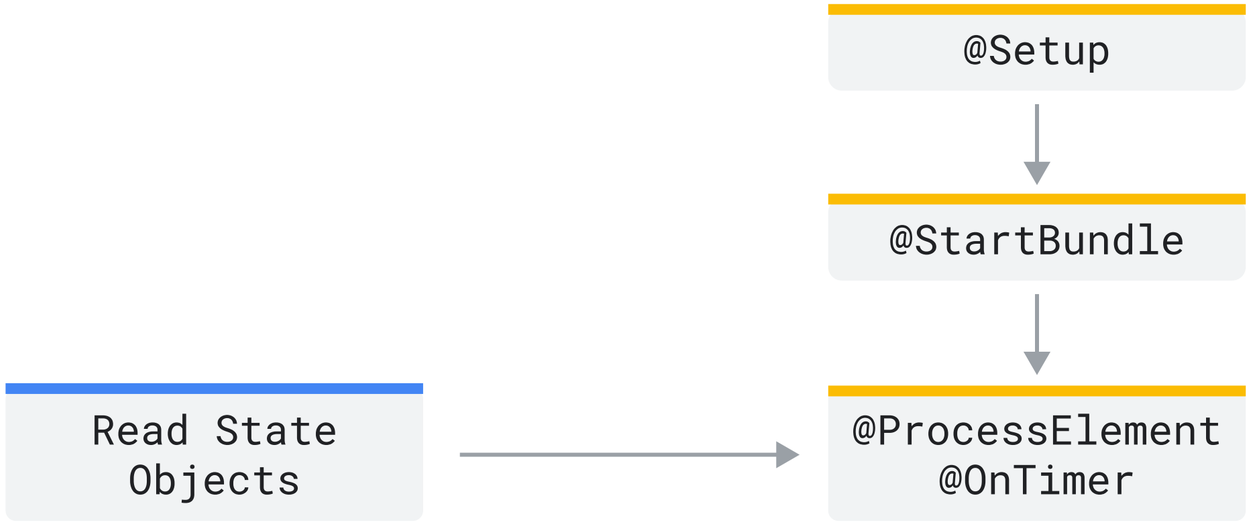
The lifecycle of a DoFn

```
@Setup
```

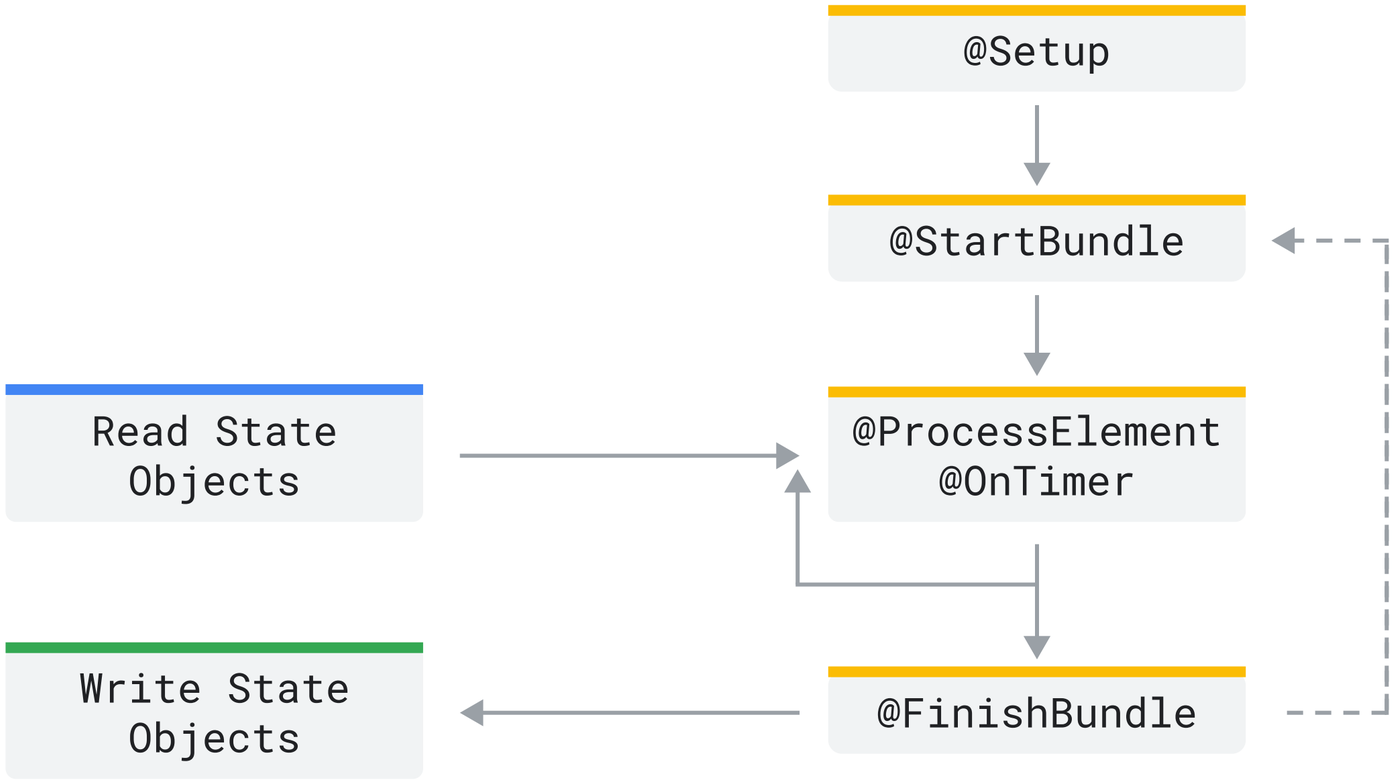
The lifecycle of a DoFn



The lifecycle of a DoFn



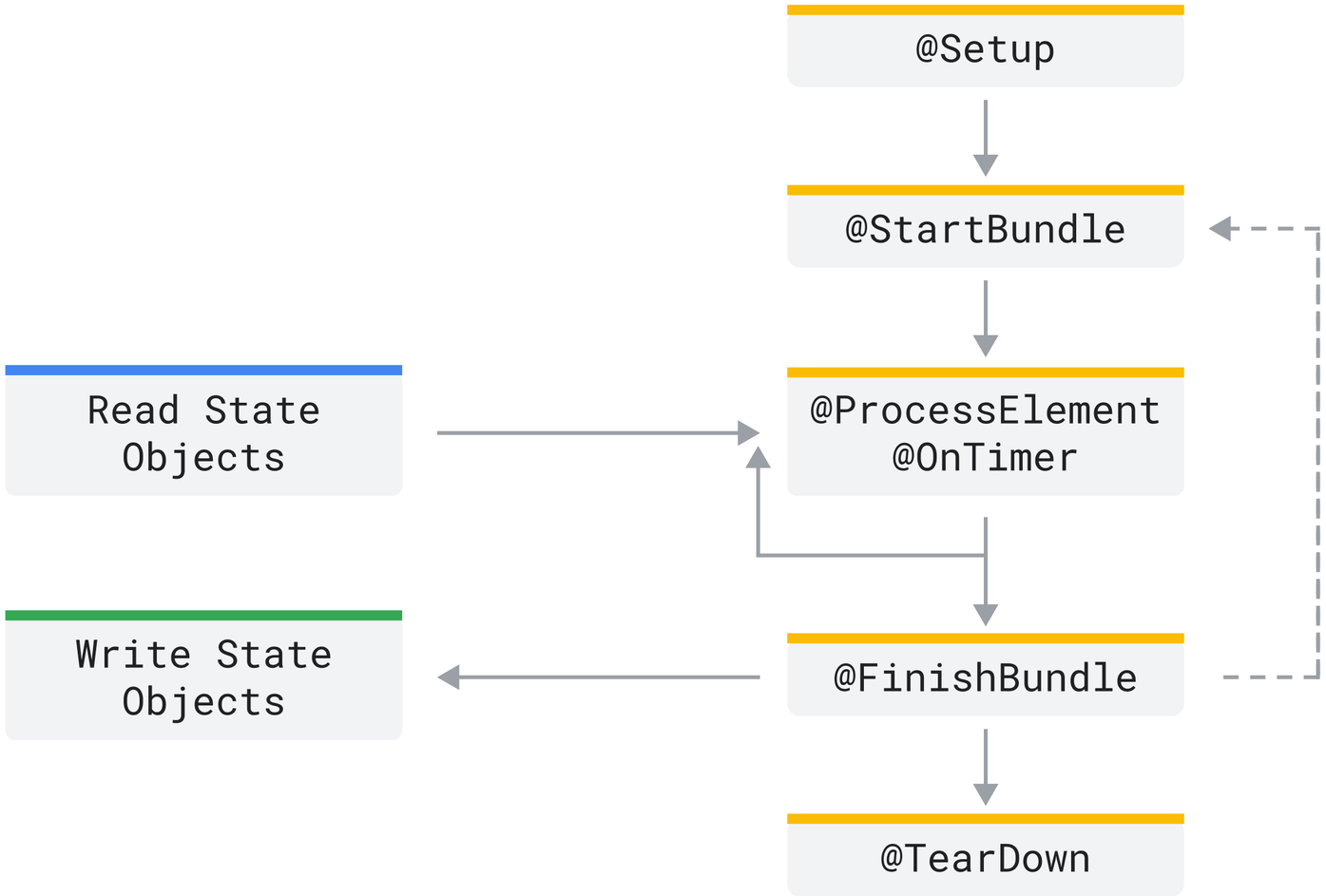
The lifecycle of a DoFn



Dependent on the runner, the DoFn object can be reused across bundles

If there is an exception at any stage @TearDown is called.

The lifecycle of a DoFn



Dependent on the runner, the DoFn object can be reused across bundles

If there is an exception at any stage @TearDown is called.