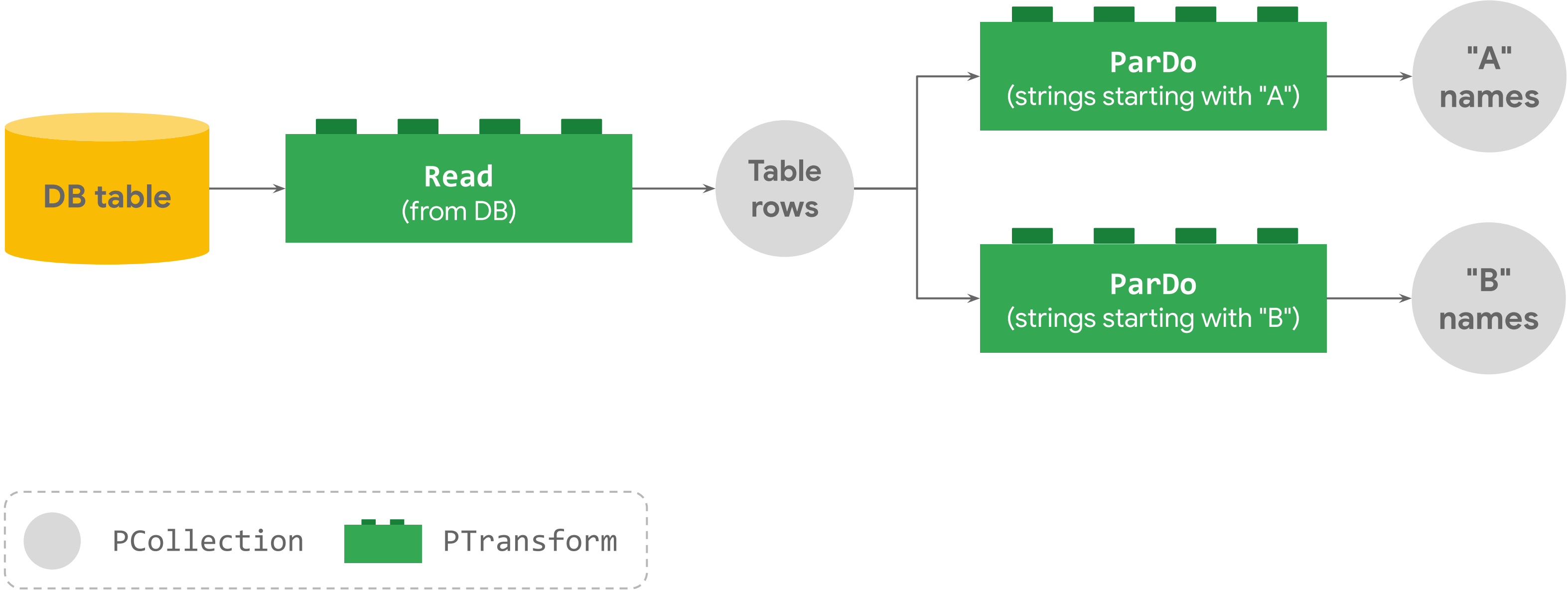
Branching & Merging PCollections

Iñigo San José
Miren Esnaola

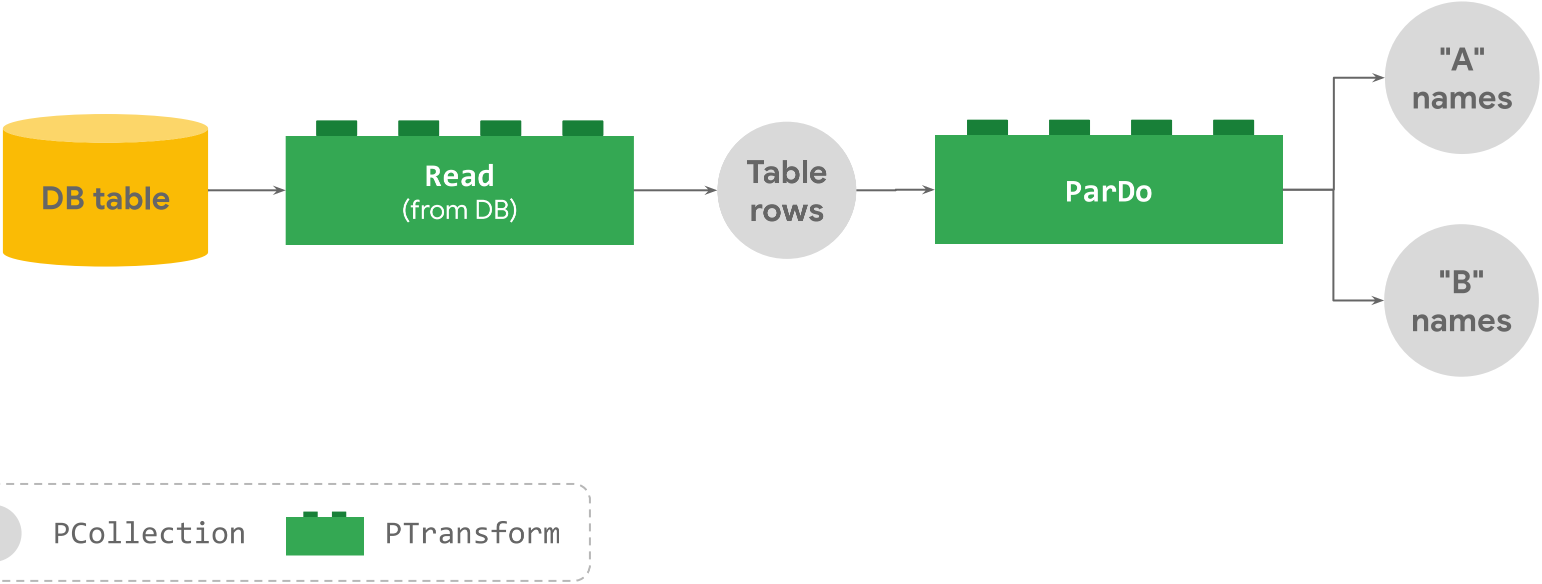
Branching PCollections

Multiple **ParDo** transforms process the same PCollection



Branching PCollections

A single **ParDo** transform producing multiple outputs



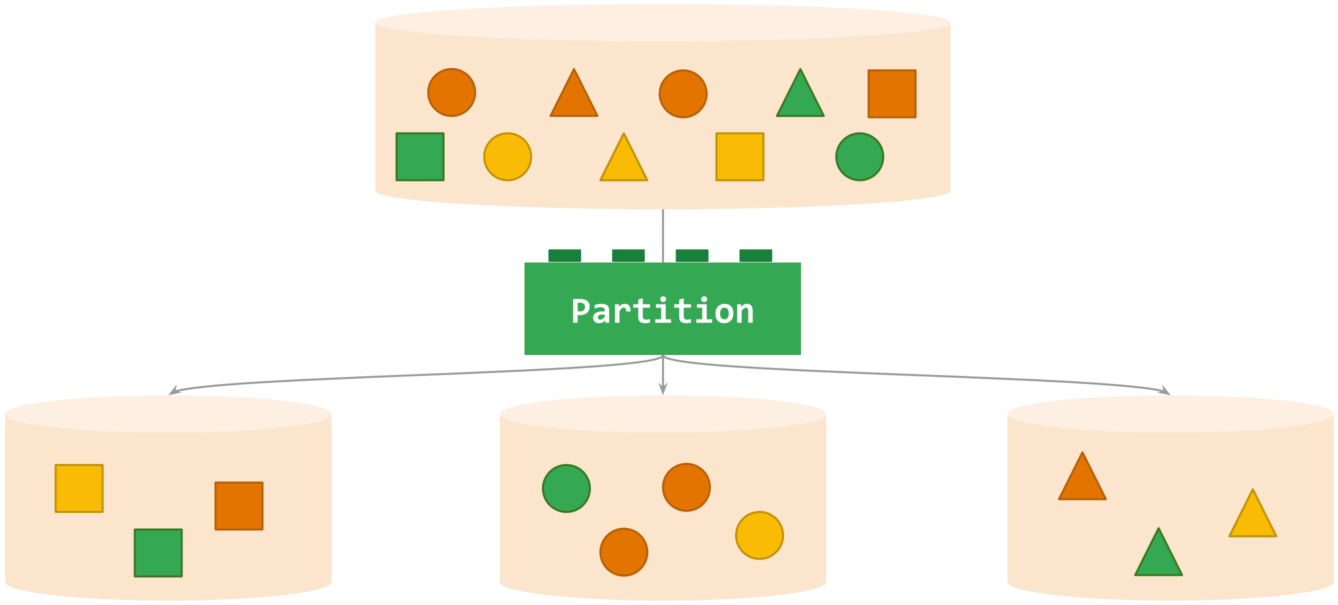
Branching `PCollections`

A **`Partition`** transform

`Partition` splits a single collection into a fixed number of smaller collections according to a partitioning function.

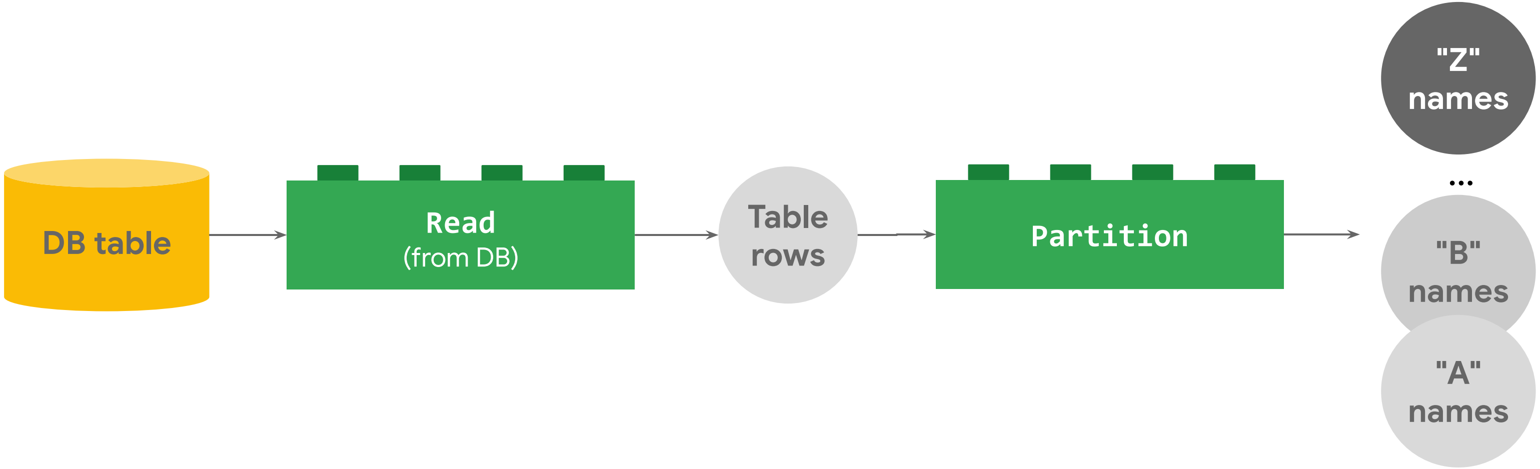
Branching PCollections

A **Partition** transform



Branching `PCollections`

A **`Partition`** transform



● PCollection 🟩 PTransform

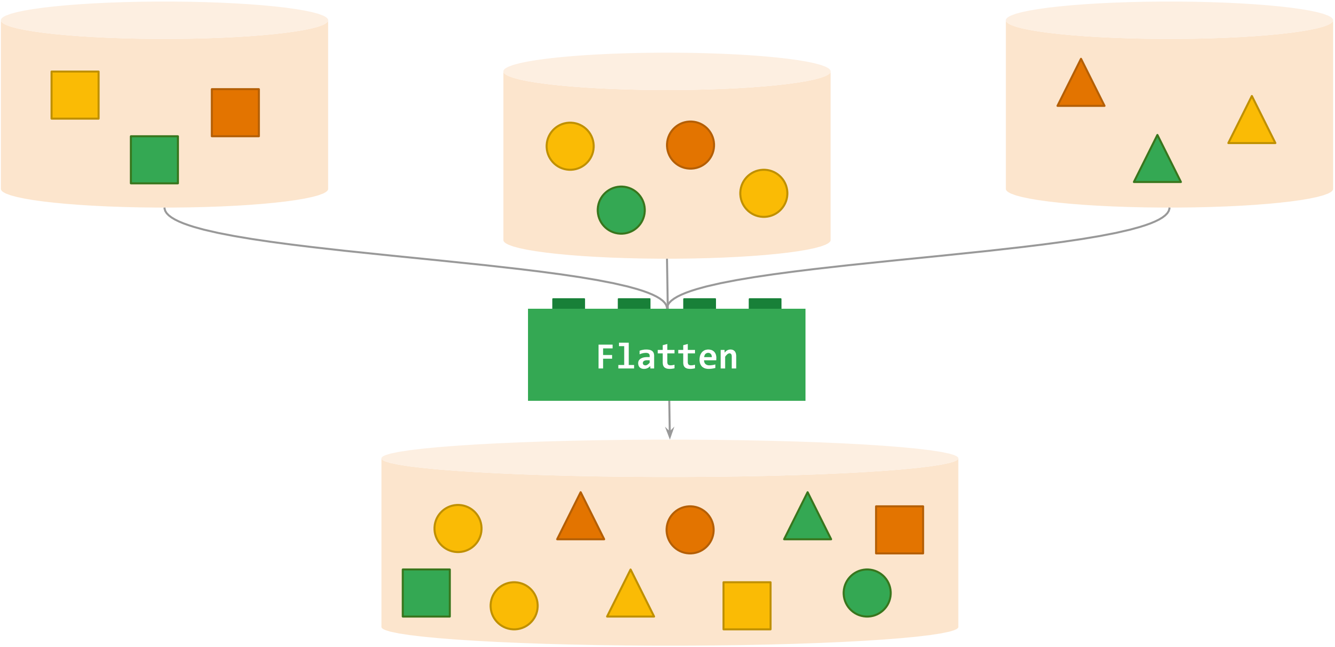
Merging `PCollections`

A **`Flatten`** transform

`Flatten` is used with collection objects that store the same data type. It merges multiple collection objects into a single logical collection.

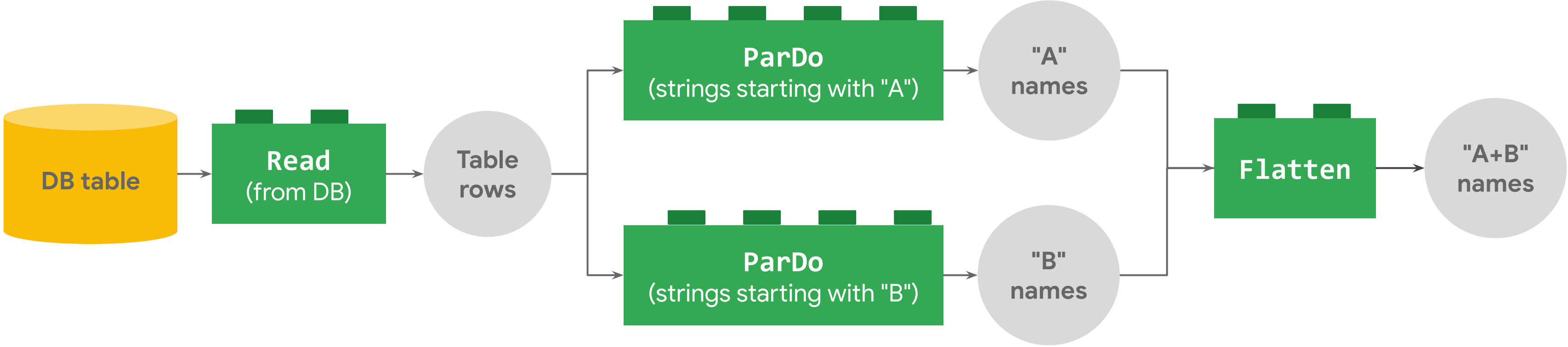
Merging PCollections

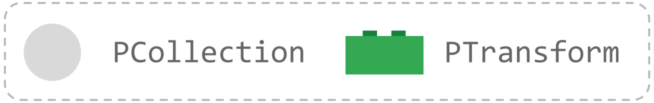
A **Flatten** transform



Merging `PCollections`

A `Flatten` transform





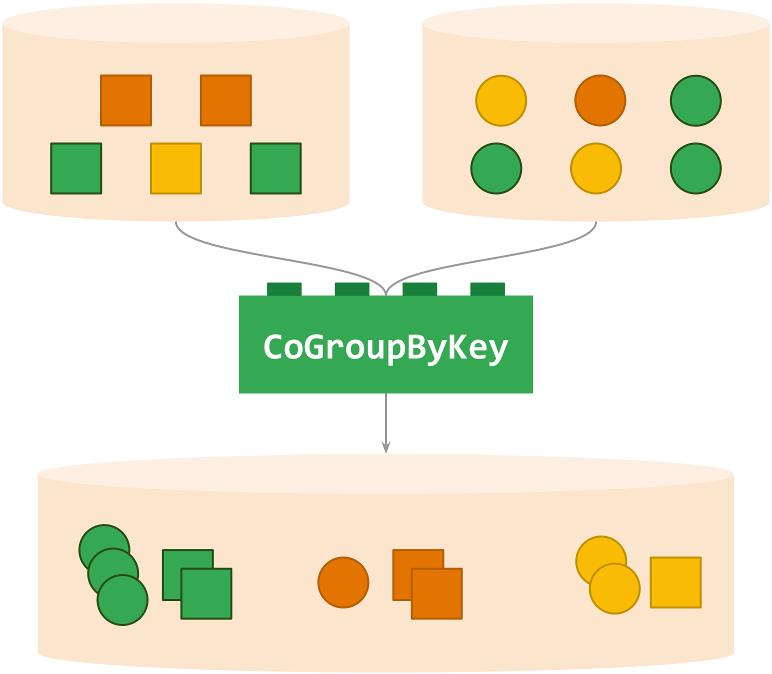
Merging `PCollections`

A **`CoGroupByKey`** transform

`CoGroupByKey` performs a relational join of two or more key/value collections with the same key type.

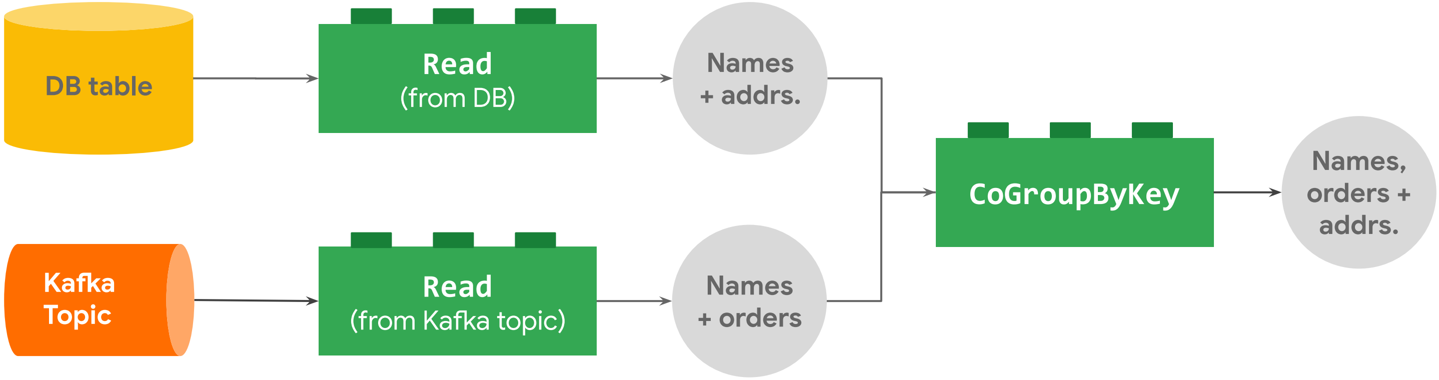
Merging `PCollections`

A **`CoGroupByKey`** transform



Merging PCollections

A CoGroupByKey transform



● PCollection 🟩 PTransform

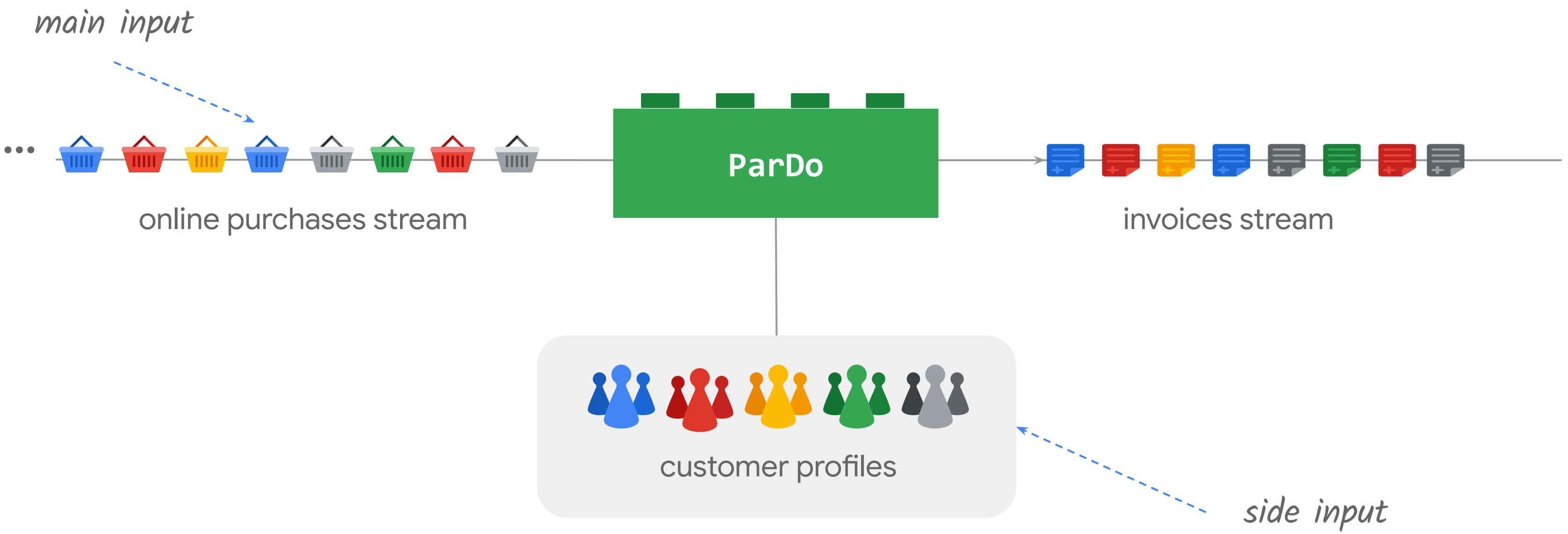
Merging `PCollections`

Side Input

A **side input** is an additional input that a `DoFn` can access each time it processes an element in the input `PCollection`. This additional data needs to be determined at runtime (it can not be hard-coded).

Merging `PCollections`

Side Input

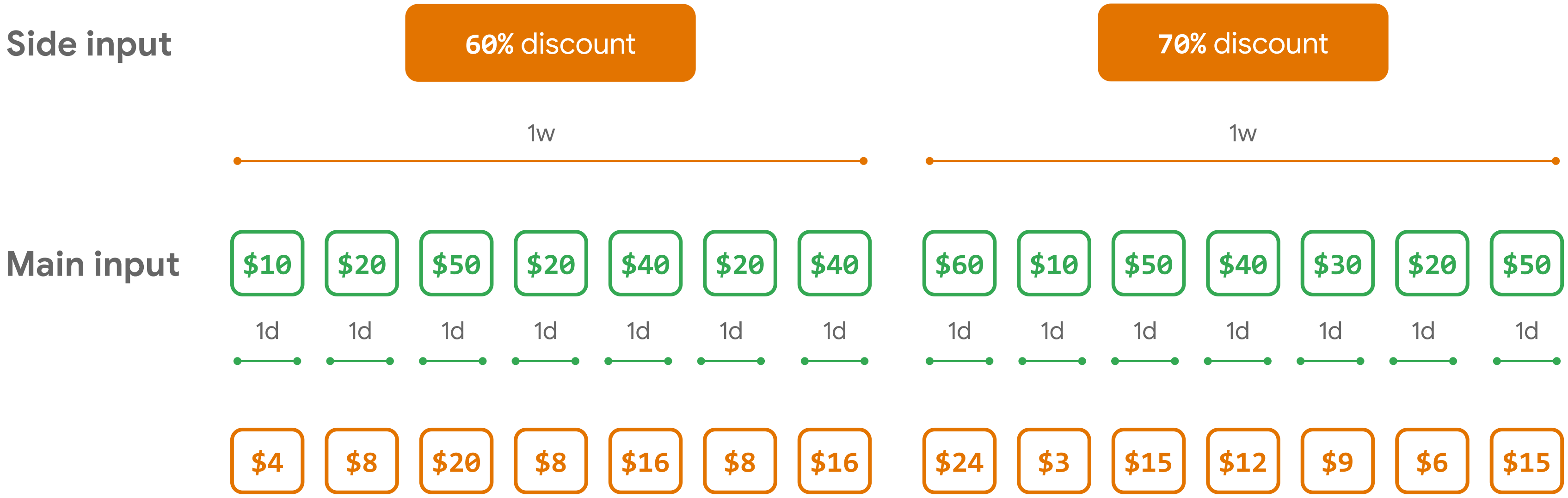


Side inputs and windowing

- When a view of a windowed `PCollection` is created, the view represents a single entity per window (one singleton per window, one list per window, ...).
- Beam projects the main input element's window into the side input's window set, and then uses the side input from the resulting window.
    - **Identical windows** → projection provides the exact corresponding window.
    - **Different windows** → projection used to choose most suitable side input window.
- If the main input element exists in more than one window, `processElement` gets called once for each window. Each call projects the "current" window for the main input element, and thus might provide a different view of the side input each time.
- If the side input has multiple trigger firings, the value from the latest trigger firing is used.

Side inputs and windowing

As the sales season goes by, the discounts are larger week by week...



Side input

60% discount

1w

70% discount

1w

Main input

$10 $20 $50 $20 $40 $20 $40 $60 $10 $50 $40 $30 $20 $50

1d 1d 1d 1d 1d 1d 1d 1d 1d 1d 1d 1d 1d 1d

$4 $8 $20 $8 $16 $8 $16 $24 $3 $15 $12 $9 $6 $15

Thank you!

Questions?

BEAM
COLLEGE