

GCP Dataflow

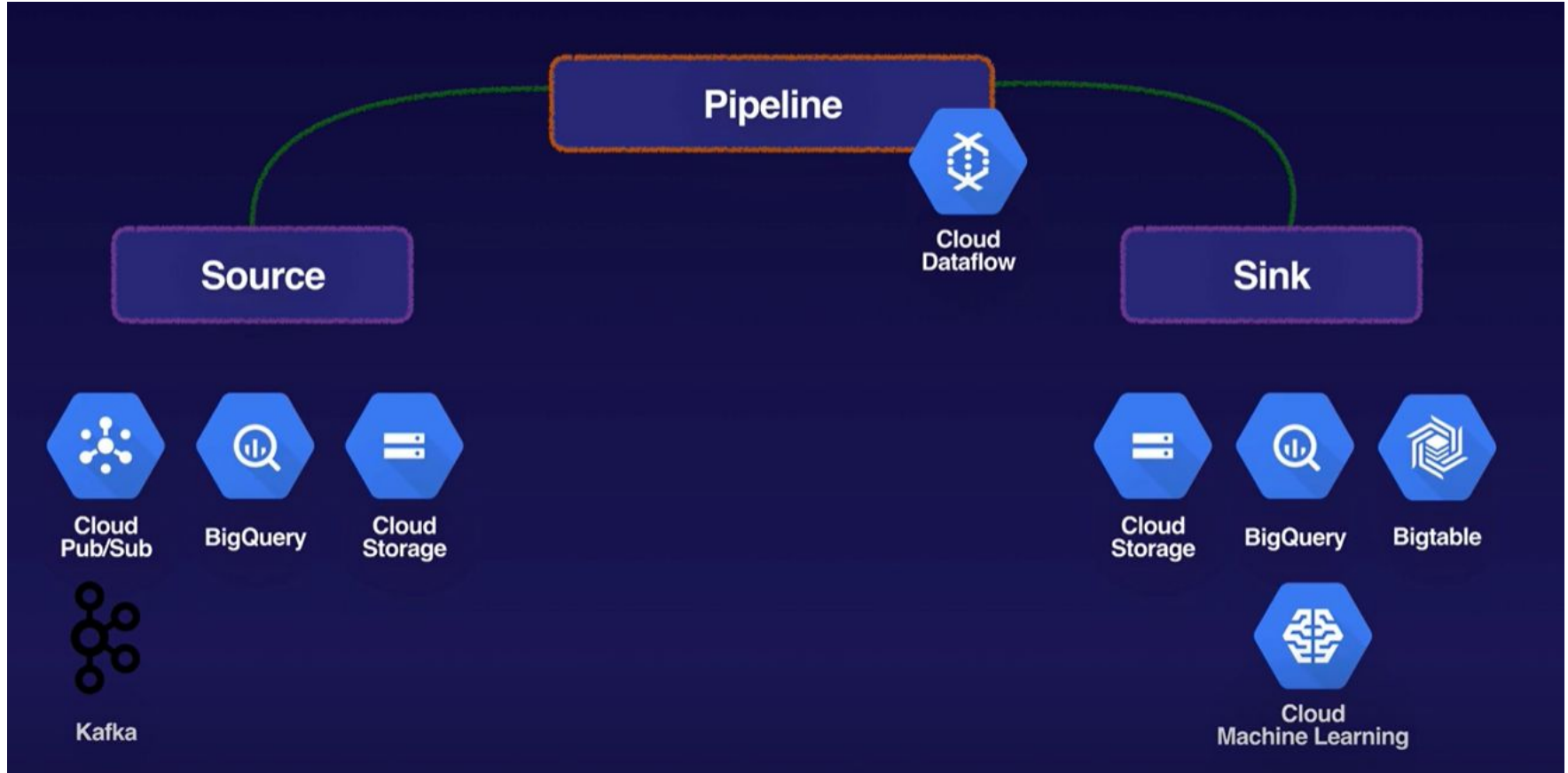
Dataflow introduction



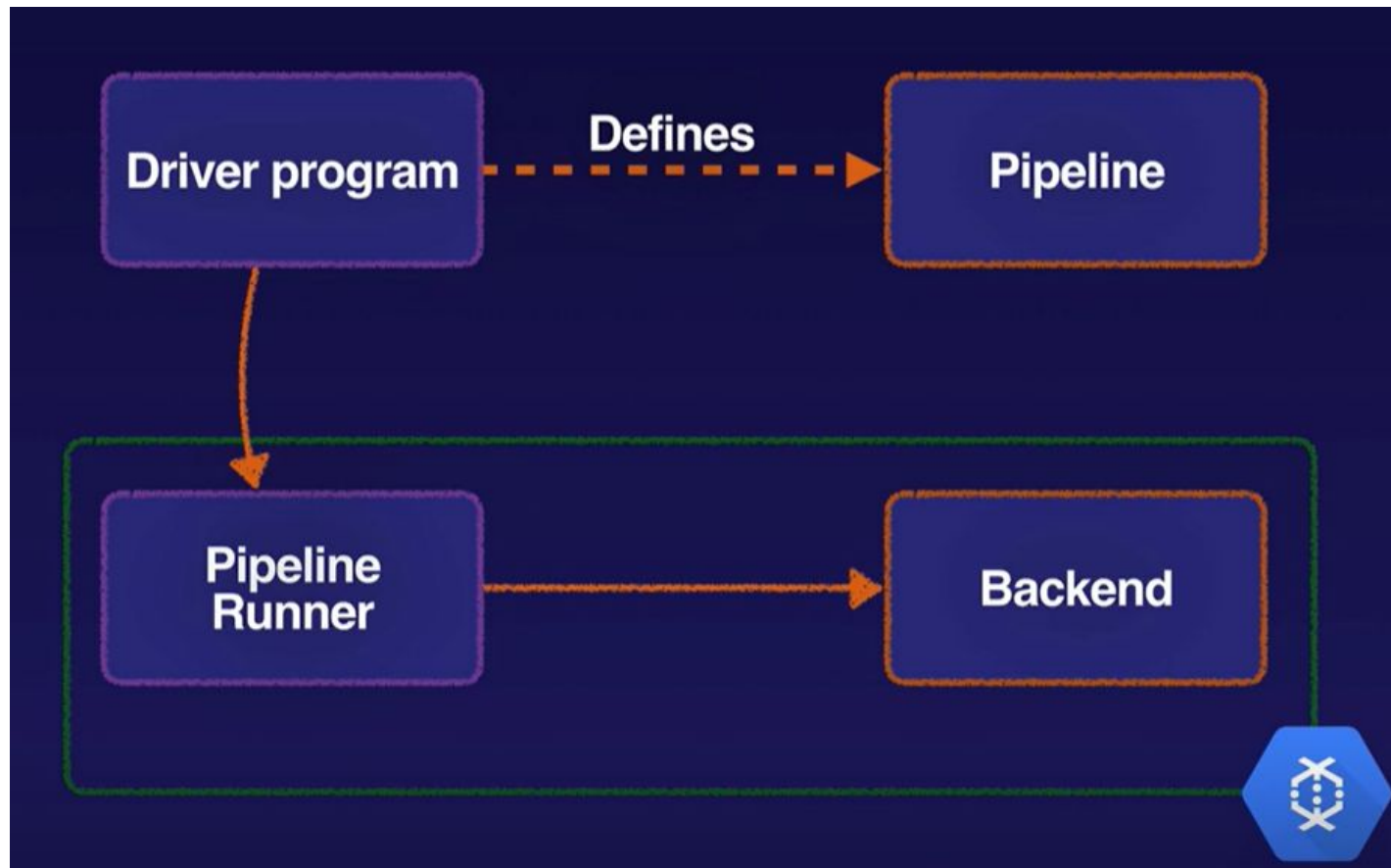
Cloud Dataflow

- Fully managed, serverless tool.
- Uses open source Apache Beam SDK.
- Supports expressive SQL, Java, and Python APIs.
- Real-time and batch processing.
- Stackdriver integration.

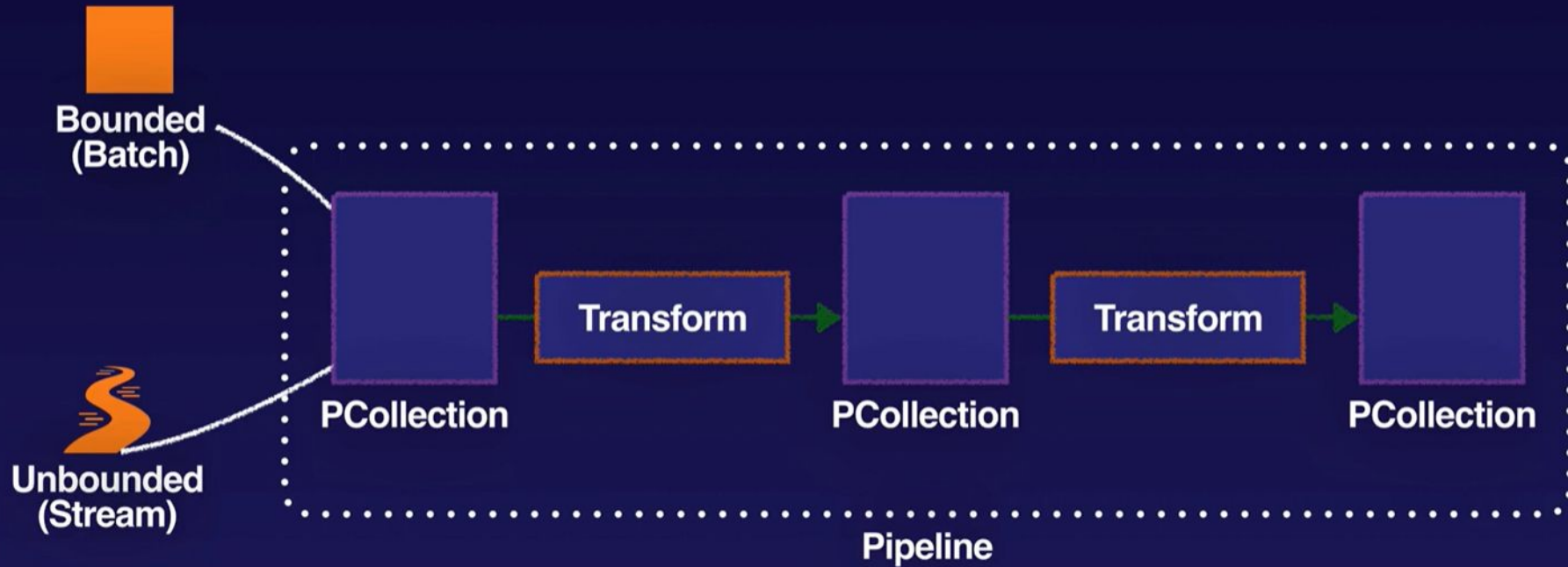
Pipeline Source and Sink



Driver Program and Runner



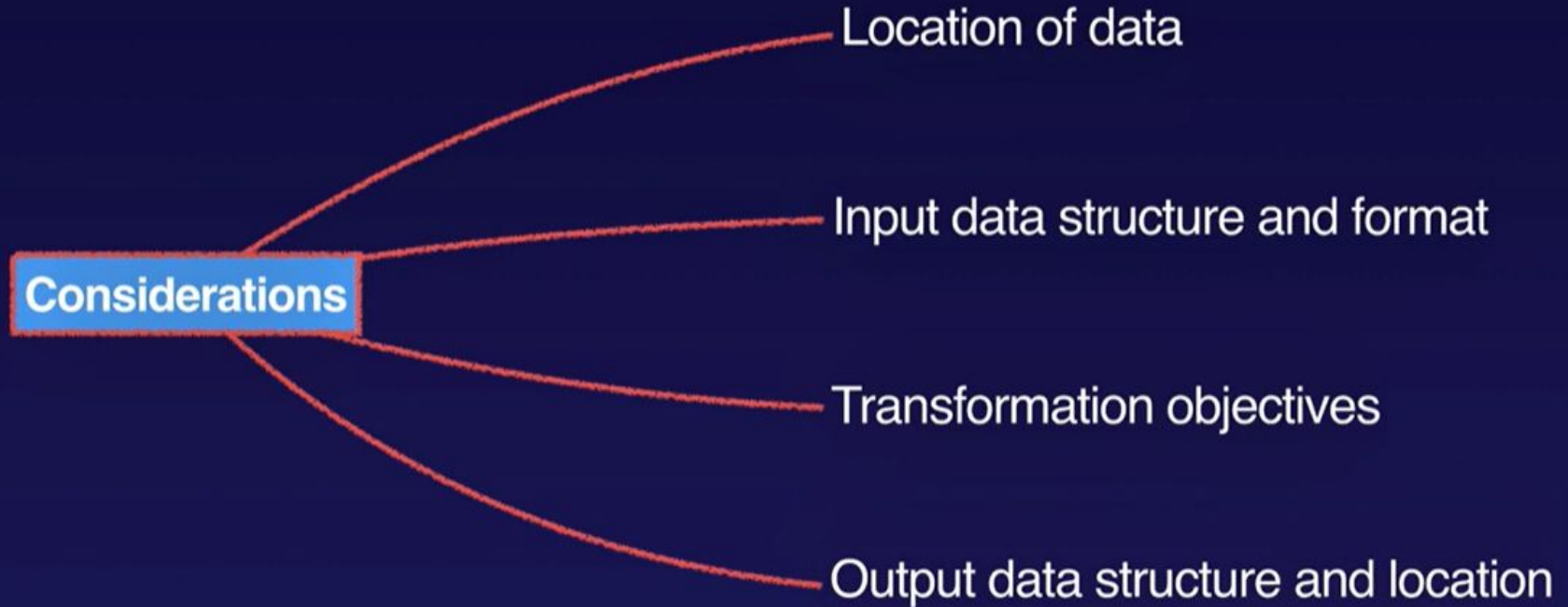
PCollections and Transforms



Pipeline Development Lifecycle



Pipeline Design Considerations

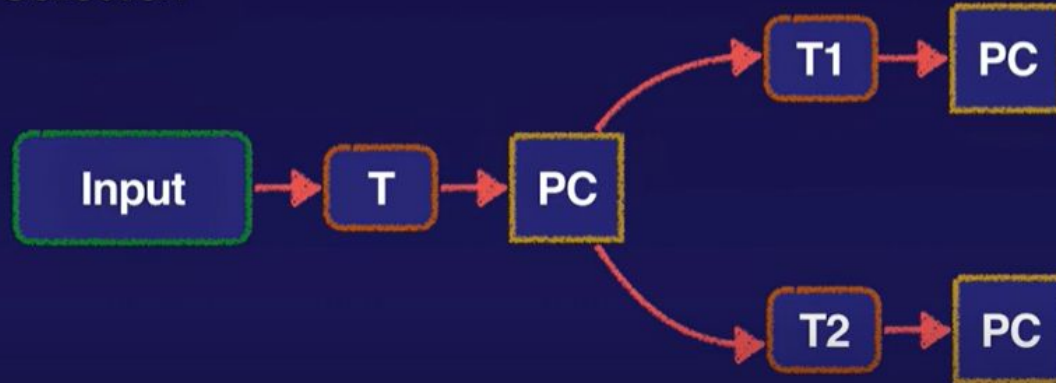


Pipeline Structure - Basic and Branching

➔ Basic pipeline

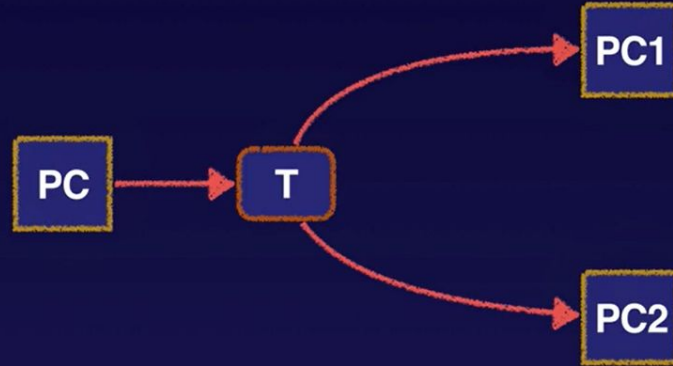


➔ Branching - PCollection

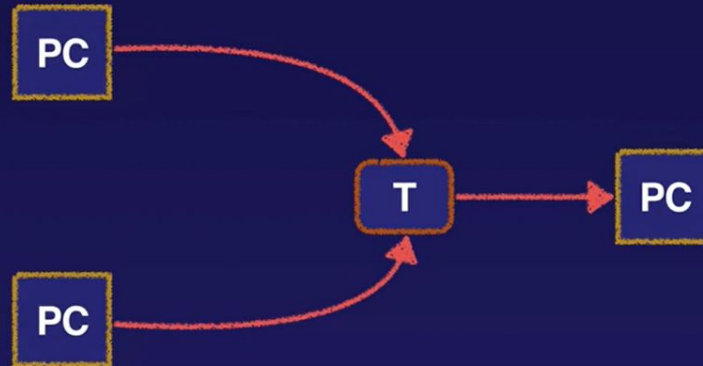


Pipeline Structure - Branching and Merging

➔ Branching - Transform

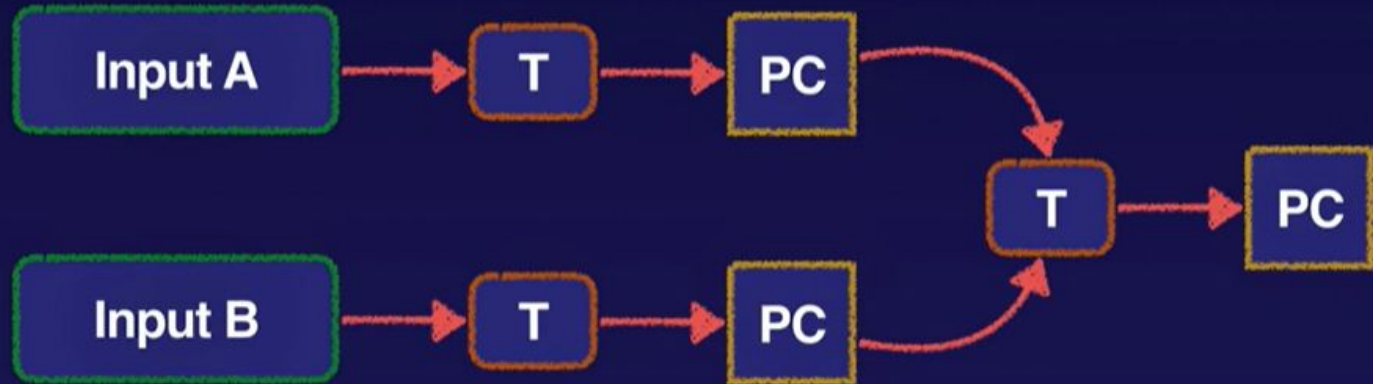


➔ Merging

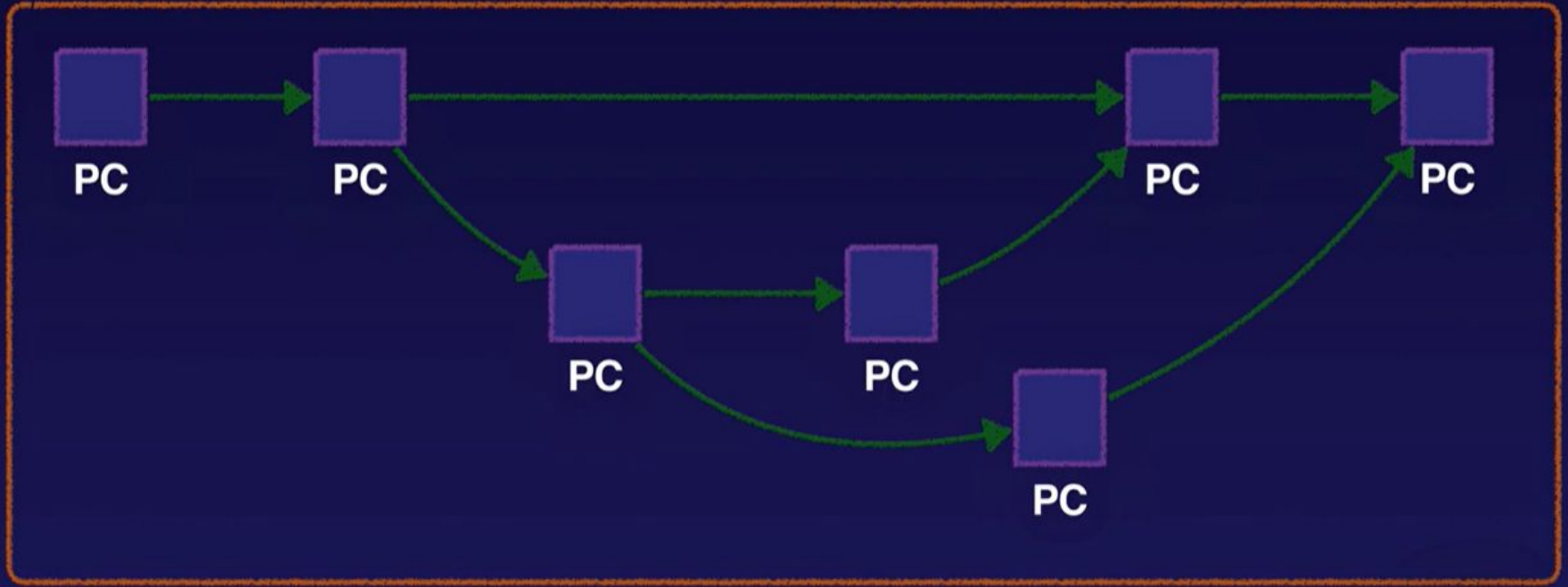


Pipeline Structure - Branching and Merging

➔ Multiple sources



Pipeline Graph - Directed Acyclic Graph



Pipeline

Pipeline Creation

Driver Program (design time)

- Create pipeline object
- Create a PCollection using read or create transform
- Apply multiple transforms as required
- Write out final PCollection

Runtime



Dataflow Pipeline Concepts

Dataflow Pipeline Concepts

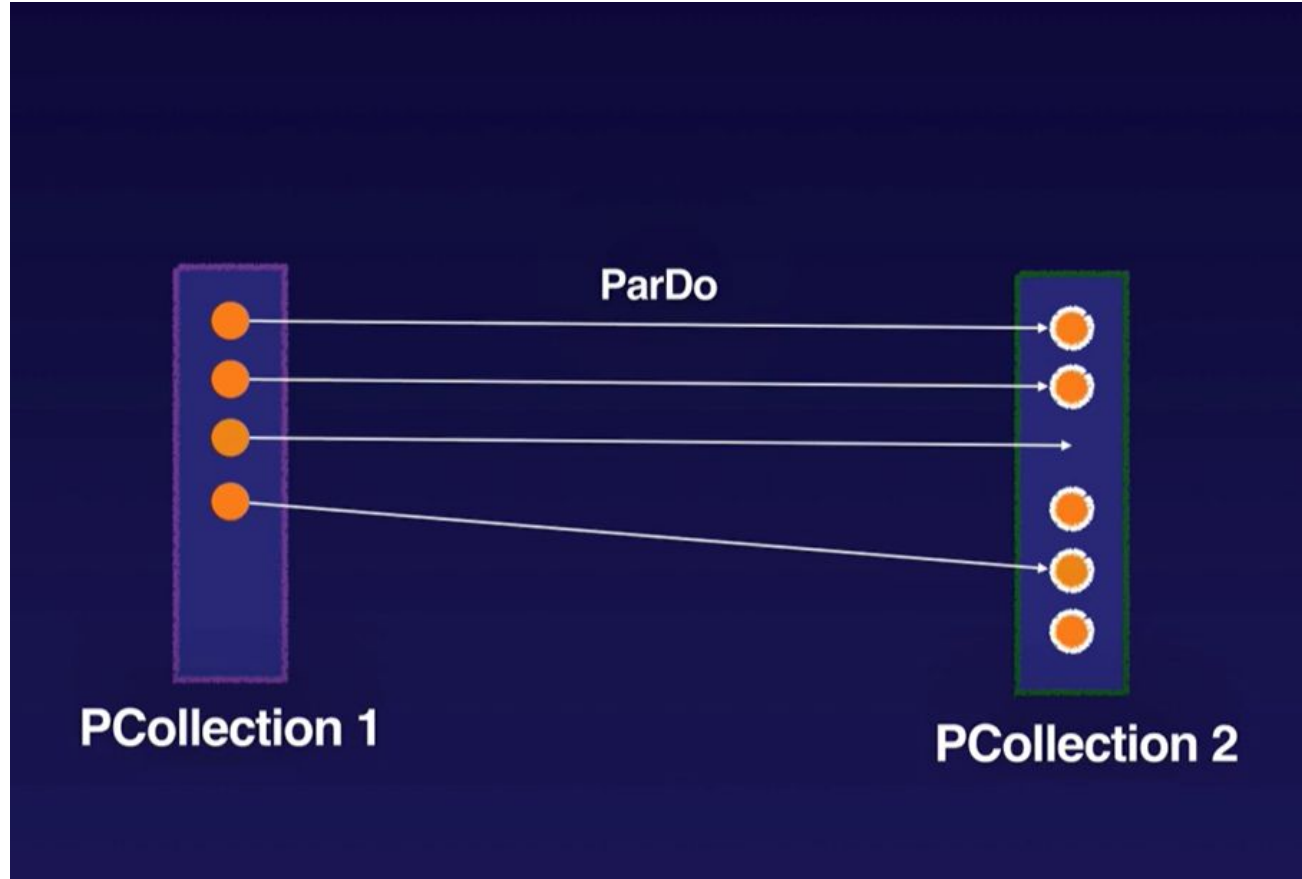
ParDo Transform

Aggregation Transforms

PCollections

Core Beam Transforms

ParDo



User-defined function (UDF)

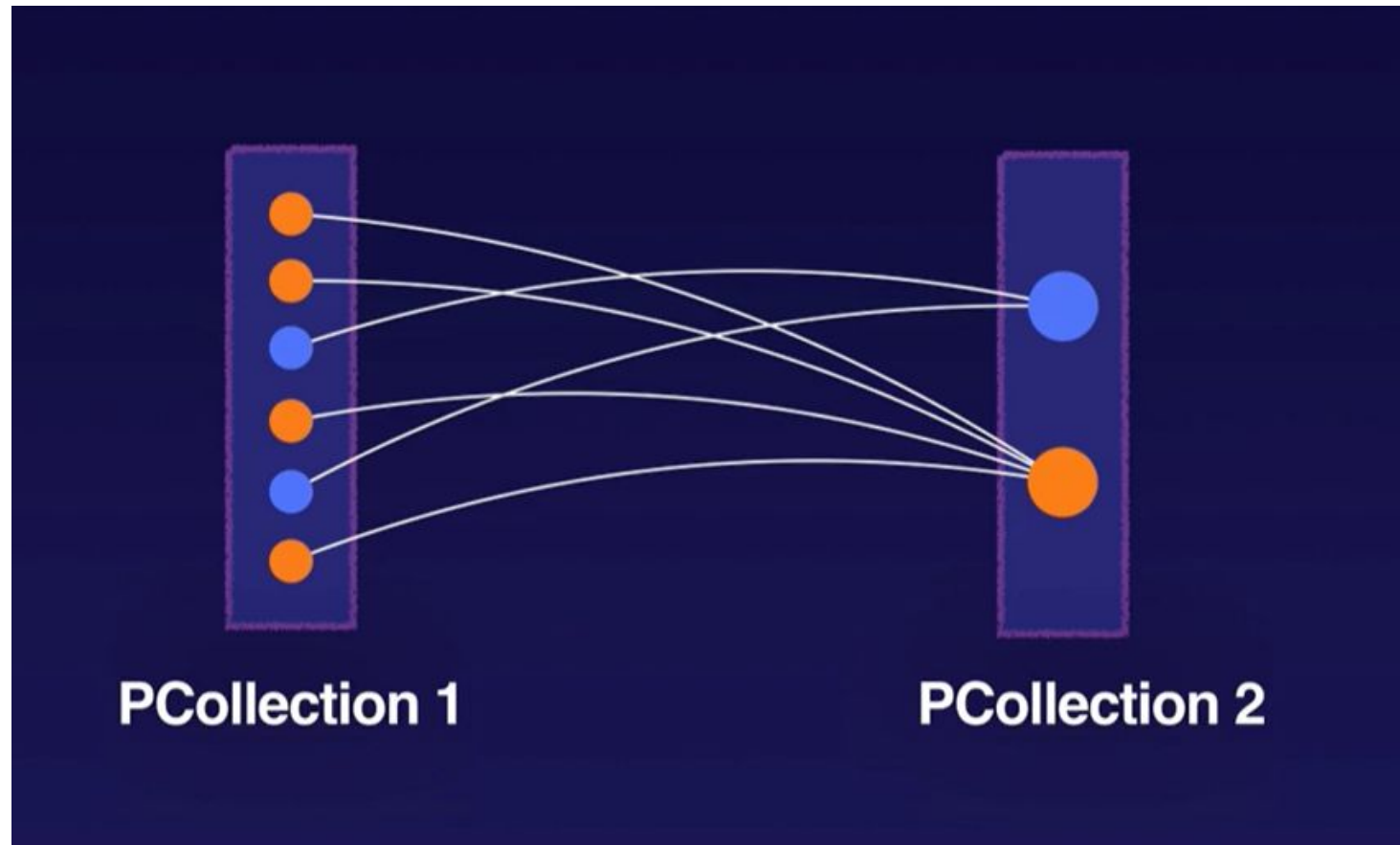
```
import apache_beam as beam
```

```
class SplitWords(beam.DoFn):  
    def __init__(self, delimiter=','):  
        self.delimiter = delimiter  
  
    def process(self, text):  
        for word in text.split(self.delimiter):  
            yield word
```

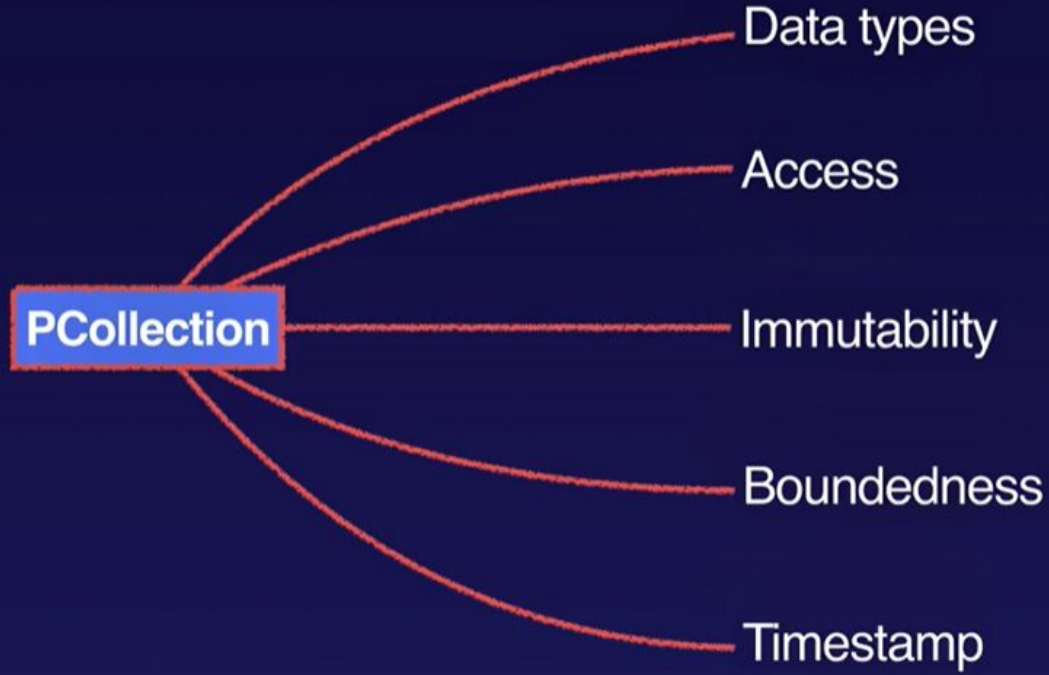
```
with beam.Pipeline() as pipeline:  
    plants = (  
        pipeline  
        | 'Sports' >> beam.Create([  
            'Rugby, Athletics, Cricket',  
            'Swimming, Waterpolo',  
        ])  
        | 'Split words' >> beam.ParDo(SplitWords(','))  
        | beam.Map(print)  
    )
```



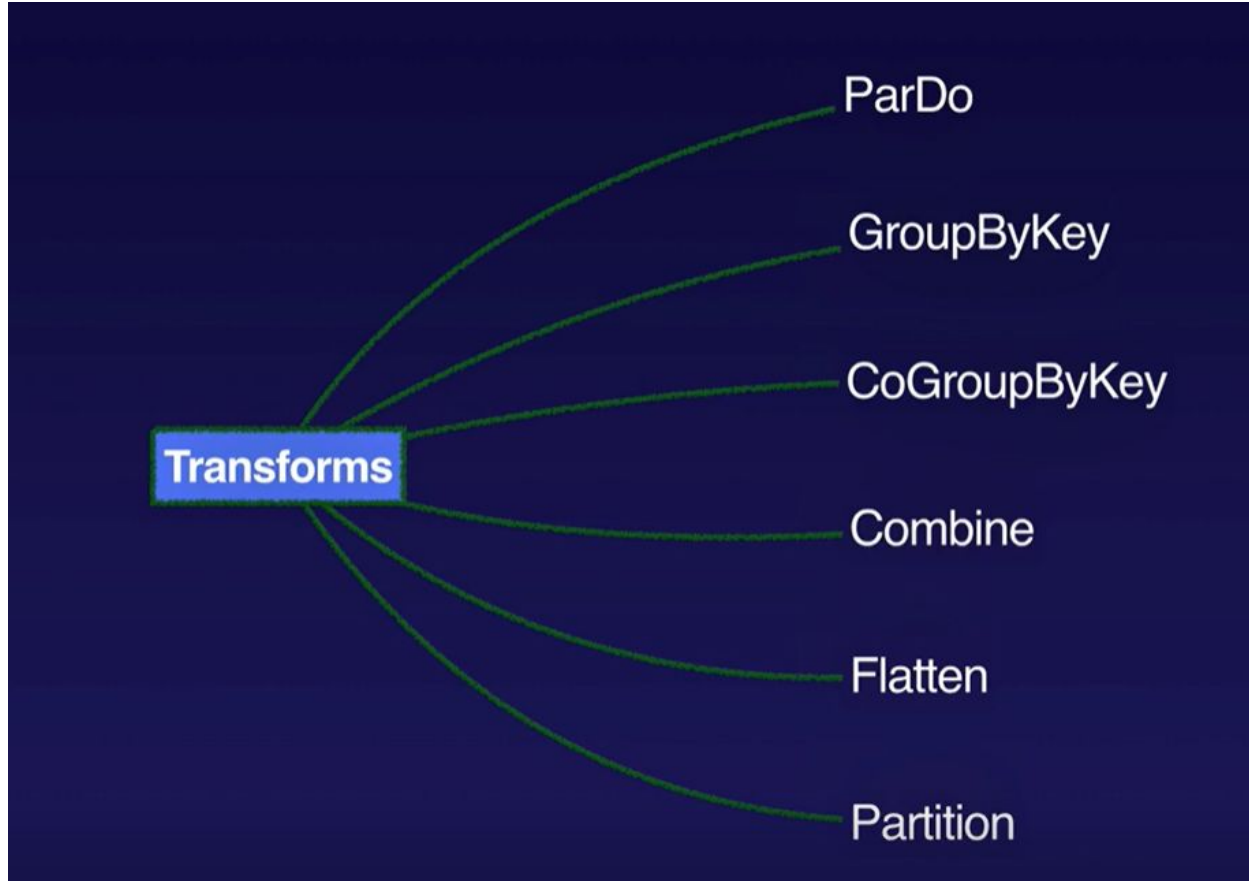
Aggregation



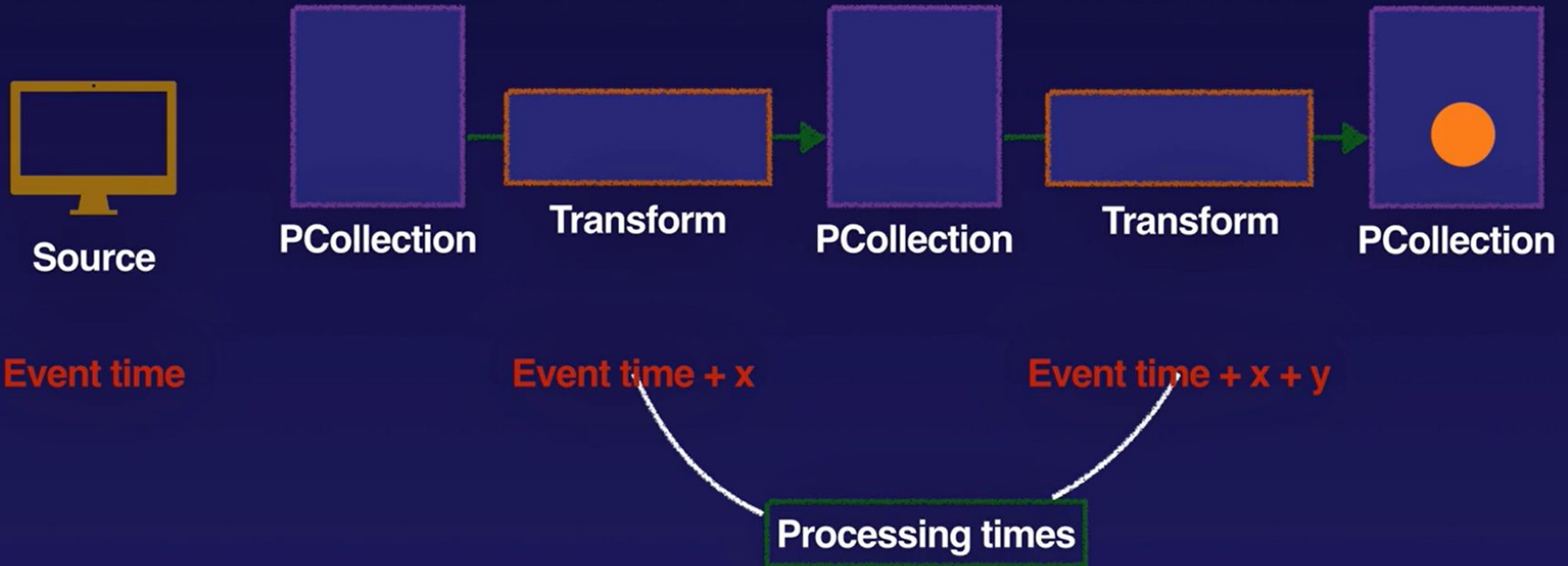
Characteristics of PCollections



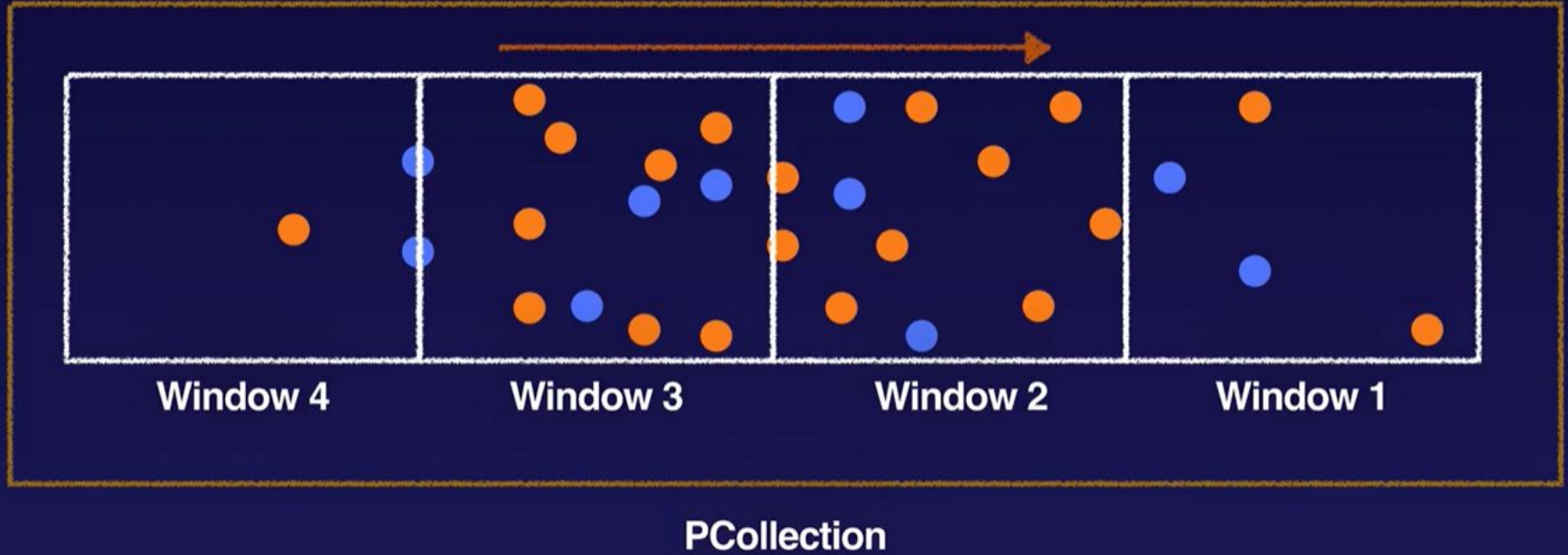
Core Beam transforms



Advanced Dataflow Concepts - Event time



Advanced Dataflow Concepts - Windowing



Advanced Dataflow Concepts

→ Fixed



→ Sliding



→ Per session



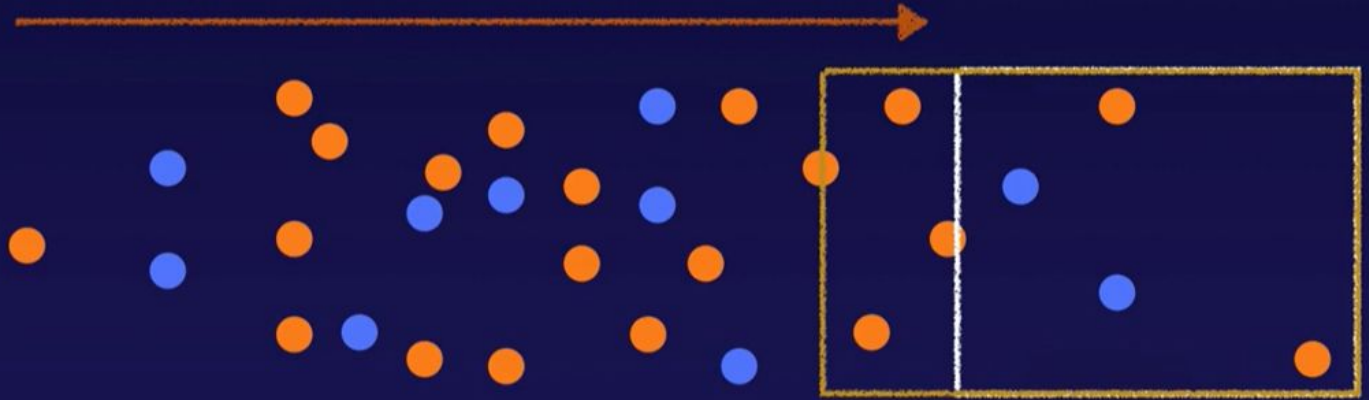
→ Single global



Watermarks



Source

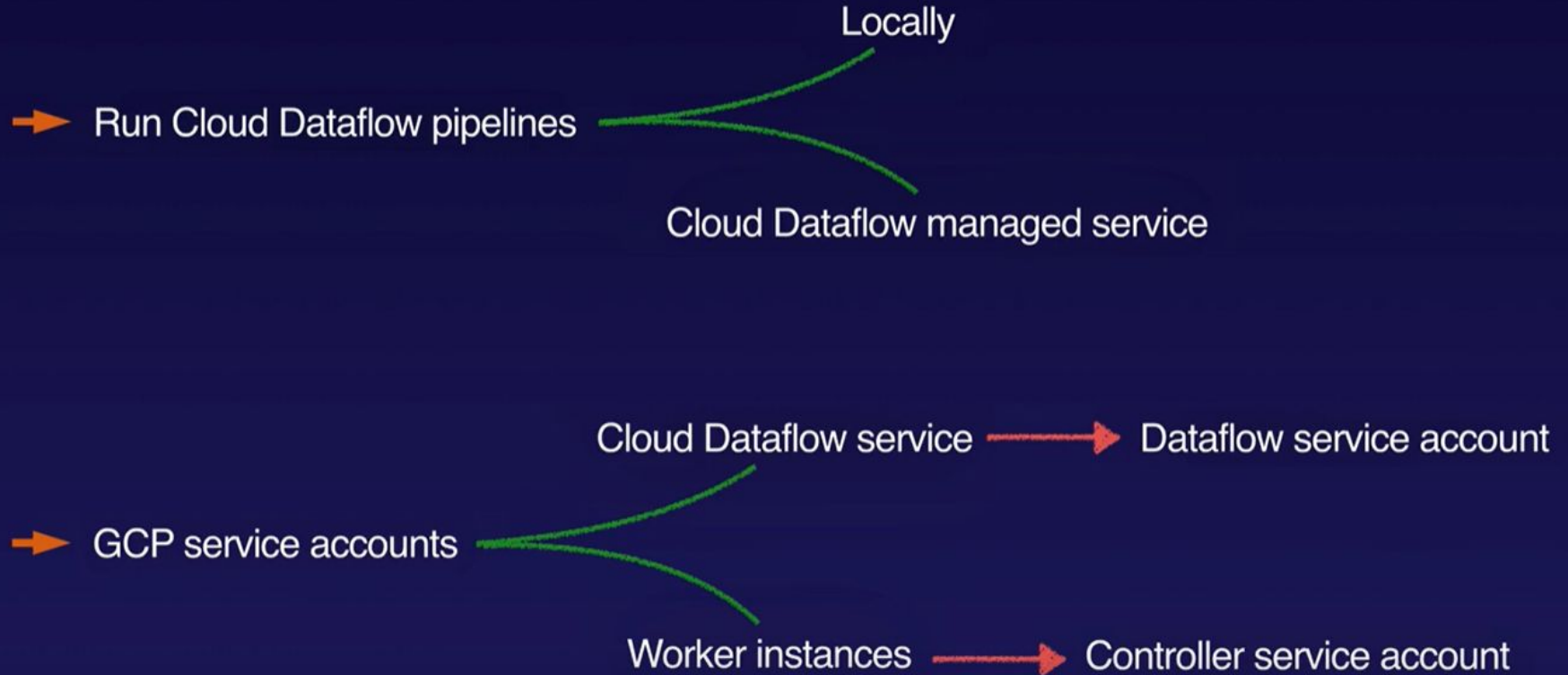


Watermark 1

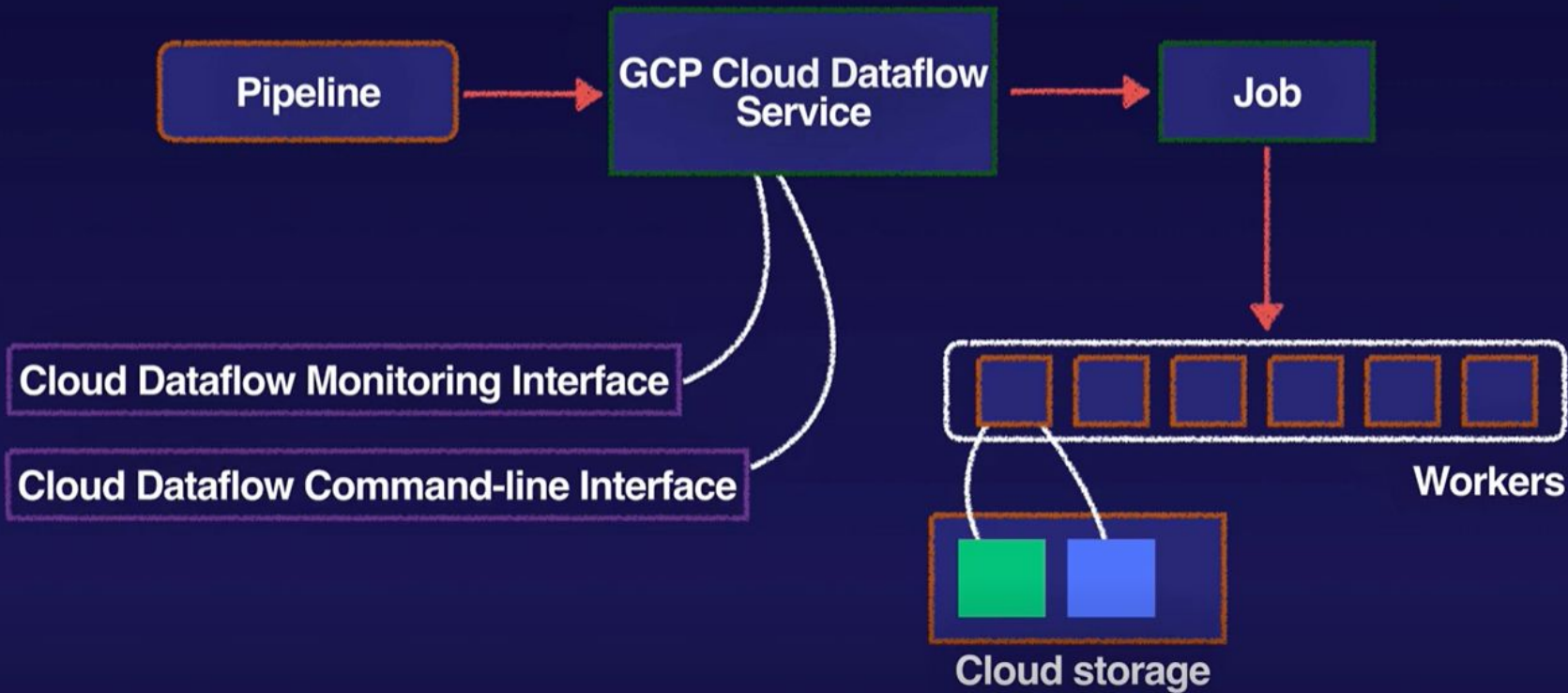
Triggers



Pipeline Access



Cloud Dataflow Managed Service



Cloud Dataflow Service Account

Cloud Dataflow service account



```
graph LR; A[Cloud Dataflow service account] --- B[Automatically created]; A --- C[Manipulates job resources]; A --- D[Cloud Dataflow service agent role]; A --- E[Read/write access to project resources];
```

Automatically created

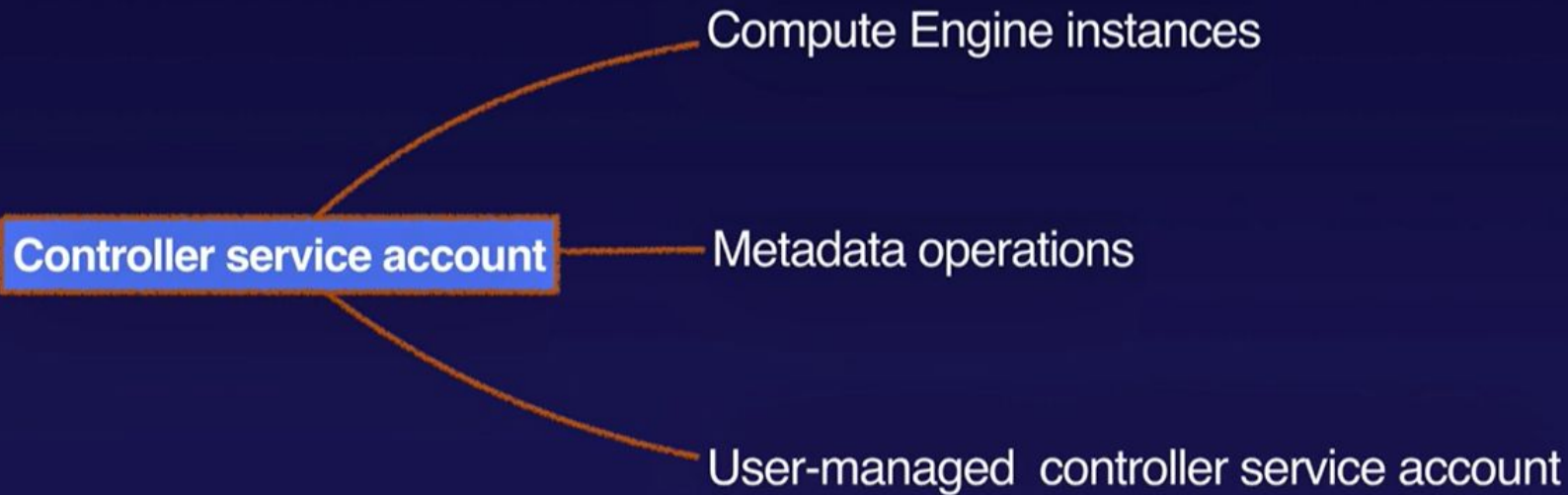
Manipulates job resources

Cloud Dataflow service agent role

Read/write access to project resources

`service-<project-number>@dataflow-service-producer-prod.iam.gserviceaccount.com`

Service Accounts



`<project-number>compute@developer.gserviceaccount.com`

Access and Security



➔ Cloud Dataflow IAM roles

Using Dataflow - Regional Endpoints

- ➔ Manages metadata about Cloud Dataflow jobs
- ➔ Controls Cloud Dataflow workers
- ➔ Automatically selects best zone

Reasons to use regional endpoints

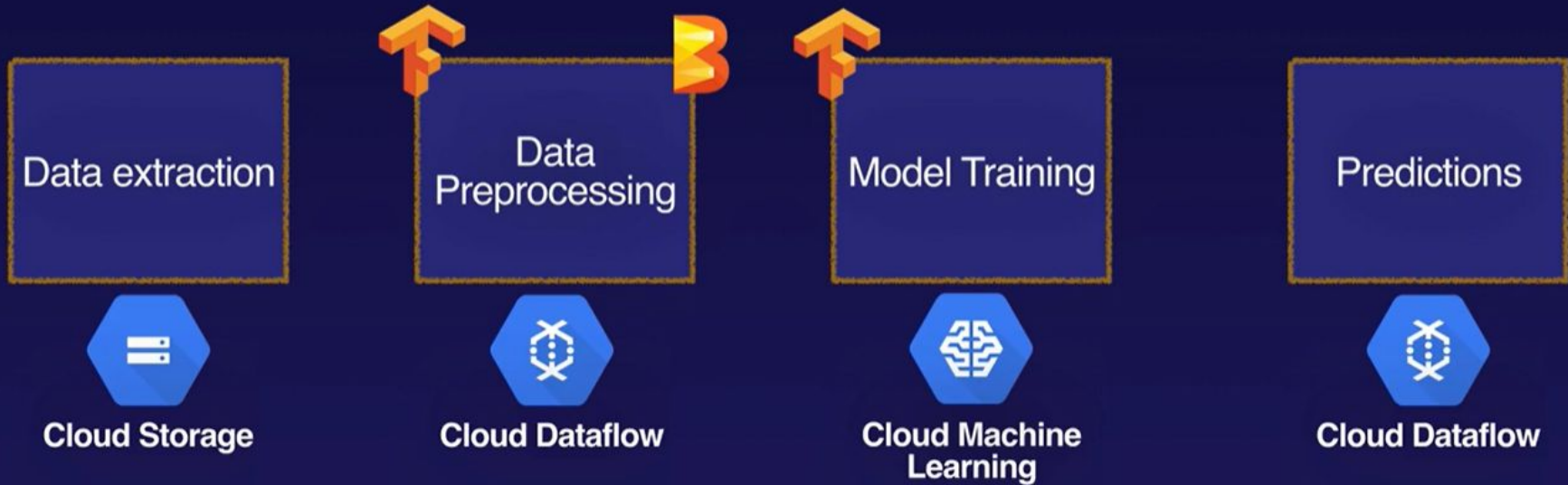
Security and compliance

Data locality

Resiliency

Machine Learning with Dataflow

➔ Apache Beam, Cloud Dataflow, TensorFlow and Cloud Machine Learning



Using Dataflow

➔ Customer-managed encryption keys

➔ Flexible Resource Scheduling (FlexRS)

Advanced scheduling

Cloud Dataflow Shuffle service

Preemptible VMs

➔ Migrating MapReduce jobs to Cloud Dataflow

➔ Cloud Dataflow with Pub/Sub Seek

Cloud Dataflow SQL

- ➔ Develop and run Cloud Dataflow jobs from the BigQuery web UI
- ➔ Cloud Dataflow SQL (ZetaSQL variant) integrates with Apache Beam SQL

Apache Beam SQL

Query bounded and unbounded PCollections

Query is converted to a SQL transform

Cloud Dataflow SQL

Utilise existing SQL skills

Join streams with BigQuery tables

Query streams or static datasets

Write output to BigQuery for analysis and visualisation